אוסף ביטויים רגולריים

מסמך זה הורד מהאתר http://underwar.livedns.co.il.
צוות האתר איננו אחראי לכל נזק, ישיר או עקיף, שיגרם עקב השימוש במידע המופיע במסמך, וכן לנכונות התוכן של הנושאים המופיעים במסמך. עם זאת, נעשו מירב המאמצים כדי לספק את המידע המדויק והמלא ביותר.
למסמך זה מחברים רבים. שם המחבר של כל אחד מהביטויים מופיע ליד הביטוי עצמו.
כל הזכויות שמורות למחברים.

ביטויים רגולריים הם אחד הכלים החזקים ביותר שבנמצא המשמשים לניתוח מחרוזות. קל יחסית לכתוב אותם, וכל שפות התכנות המודרניות מכילות ספריות התומכות בשימוש בביטויים אלו.
בעזרת ביטויים רגולריים נוכל לזהות במהירות האם מחרוזת שבידינו היא מספר, מילה, או אולי כתובת אימייל, והאפשרויות הן רבות.
למרות ששימוש בביטויים רגולריים הוא לעיתים איטי יותר מאלגוריתם שיפעל על המערך המכיל את המחרוזת בצורה ישירה, נעדיף לרוב להשתמש בביטויים רגולים, וזאת ממספר סיבות:

- אלגוריתם המנתח מחרוזת צורך זמן בדיקה רב, ובאגים עלולים להתחבא בו. לעומת שימוש באלגוריתמים, ביטויים רגולריים ניתנים להוכחה ולבדיקה בקלות רבה יותר.
- ביטויים רגולריים ניתנים לשינוי במהירות, בעוד שאלגוריתם המותאם לבעיה ידרוש שינוי (לפעמים שינוי כולל) במידה ומבנה המחרוזת המצופה ישתנה.

במסמך זה אספנו ביטויים רגולריים שנשלחו על ידי גולשים רבים ברשת, ושניתן להשתמש בהם "כמו שהם".
בנוסף העלנו לאתר UnderWarrior Project מסמך נוסף המלמד כיצד לכתוב ביטויים רגולריים.
מומלץ לשלב את קריאת מסמכים אלו. כתיבת ביטויים רגולריים איננה מלאכה מסובכת - אך עם זאת, קריאה של דוגמאות תמיד עוזרת להשיג הבנה טובה יותר. כמו כן, אין צורך "להמציא את הגלגל". סביר להניח שביטויים רגולריים רבים מופיעים בתוכניות שונות. במקום לפתח את הביטויים הרגולריים, מסמך זה יכול לשמש מעין מילון קטן לביטויים אלו.

Expression:	**^[a-zA-Z]{1,2}[0-9][0-9A-Za-z]{0,1} {0,1}[0-9][A-Za-z]{2}$**
Description:	Matches UK postcodes according to the following rules 1. LN NLL eg N1 1AA 2. LLN NLL eg SW4 0QL 3. LNN NLL eg M23 4PJ 4. LLNN NLL eg WS14 0JT 5. LLNL NLL eg SW1N 4TB 6. LNL NLL eg W1C 8LQ Thanks to Simon Bell for information ...
Matches:	[G1 1AA], [EH10 2QQ], [SW1 1ZZ]
Non-Matches:	[G111 1AA], [X10 WW], [DDD 5WW]
Submitted By:	Dave Sparks

Expression:	**^[\w-]+(?:\.[\w-]+)*@(?:[\w-]+\.)+[a-zA-Z]{2,7}$**
Description:	Matches 99.99% of e-mail addresses (excludes IP e-mails, which are rarely used). The {2,7} at the end leaves space for top level domains as short as .ca but leaves room for new ones like .museum, etc. The ?: notation is a perl non-capturing notation, a ...
Matches:	[joe.tillis@unit.army.mil], [jack_rabbit@slims.com], [foo99@foo.co.uk]
Non-Matches:	[find_the_mistake.@foo.org], [.prefix.@some.net]
Submitted By:	J. Washam

Expression:	**^([a-zA-Z0-9_\-\.]+)@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([a-zA-Z0-9\-]+\.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)$**
Description:	Email validator that adheres directly to the specification for email address naming. It allows for everything from ipaddress and country-code domains, to very rare characters in the username.
Matches:	[asmith@mactec.com], [foo12@foo.edu], [bob.smith@foo.tv]
Non-Matches:	[joe], [@foo.com], [a@a]
Submitted By:	Andy Smith

Expression:	**^\d{1,2}\/\d{1,2}\/\d{4}$**
Description:	This regular expression matches dates of the form XX/XX/YYYY where XX can be 1 or 2 digits long and YYYY is always 4 digits long.
Matches:	[4/1/2001], [12/12/2001], [55/5/3434]
Non-Matches:	[1/1/01], [12 Jan 01], [1-1-2001]
Submitted By:	Steven Smith

Expression:	**^(?:(?:(?:0?[13578]|1[02])(\/|-|\.)31)\1|(?:(?:0?[1,3-9]|1[0-2])(\/|-|\.)(?:29|30)\2))(?:(?:1[6-9]|[2-9]\d)?\d{2})$\|^(?:0?2(\/\|-\|\.)29\3(?:(?:(?:1[6-9]\|[2-9]\d)?(?:0[48]\|[2468][048]\|[13579][26])\|(?:(?:16\|[2468][048]\|[3579][26])00))))$|^(?:(?:0?[1-9])|(?:1[0-2]))(\/|-|\.)(?:0?[1-9]|1\d|2[0-8])\4(?:(?:1[6-9]|[2-9]\d)?\d{2})$**
Description:	This expression validates dates in the US m/d/y format from 1/1/1600 - 12/31/9999. The days are validated for the given month and year. Leap years are validated for all 4 digits years from 1600-9999, and all 2 digits years except 00 since it could be a ...
Matches:	[01.1.02], [11-30-2001], [2/29/2000]

Non-Matches:	[02/29/01], [13/01/2002], [11/00/02]
Submitted By:	Michael Ash

Expression:	**^(25[0-5]|2[0-4][0-9]|[0-1]{1}[0-9]{2}|[1-9]{1}[0-9]{1}|[1-9])\.(25[0-5]|2[0-4][0-9]|[0-1]{1}[0-9]{2}|[1-9]{1}[0-9]{1}|[1-9]|0)\.(25[0-5]|2[0-4][0-9]|[0-1]{1}[0-9]{2}|[1-9]{1}[0-9]{1}|[1-9]|0)\.(25[0-5]|2[0-4][0-9]|[0-1]{1}[0-9]{2}|[1-9]{1}[0-9]{1}|[0-9])$**
Description:	RegExp for validating the format of IP Addresses. This works great with the ASP.NET RegularExpressionValidator server control.
Matches:	[127.0.0.1], [255.255.255.0], [192.168.0.1]
Non-Matches:	[1200.5.4.3], [abc.def.ghi.jkl], [255.foo.bar.1]
Submitted By:	G. Andrew Duthie

Expression:	**(AUX|PRN|NUL|COM\d|LPT\d)+\s*$**
Description:	"Be careful when opening or creating files by using Scripting File System Object. If the filename is based on the user's input, the user might attempt to open a serial port or printer."
Matches:	[COM1], [AUX], [LPT1]
Non-Matches:	[image.jpg], [index.html], [readme.txt]
Submitted By:	Chris Craft

Expression:	**^(?:(?:31(\/|-|\.)(?:0?[13578]|1[02]))\1|(?:(?:29|30)(\/|-|\.)(?:0?[1,3-9]|1[0-2])\2))(?:(?:1[6-9]|[2-9]\d)?\d{2})$\|^(?:29(\/\|-\|\.)0?2\3(?:(?:(?:1[6-9]\|[2-9]\d)?(?:0[48]\|[2468][048]\|[13579][26])\|(?:(?:16\|[2468][048]\|[3579][26])00))))$|^(?:0?[1-9]|1\d|2[0-8])(\/|-|\.)(?:(?:0?[1-9])|(?:1[0-2]))\4(?:(?:1[6-9]|[2-9]\d)?\d{2})$**
Description:	This expression validates dates in the ITALIAN d/m/y format from 1/1/1600 - 31/12/9999. The days are validated for the given month and year. Leap years are validated for all 4 digits years from 1600-9999, and all 2 digits years except 00 since it could b ...
Matches:	[29/02/1972], [5-9-98], [10-11-2002]
Non-Matches:	[29/02/2003], [12/13/2002], [1-1-1500]
Submitted By:	Marco Storti

Expression:	**^(user=([a-z0-9]+,)*([a-z0-9]+){1});)?(group=([a-z0-9]+,)*([a-z0-9]+){1});)?(level=[0-9]+;)?$**
Description:	This re was used for a security routine. The format is: [user=name1,name2,...,nameN;][group=group1,group2,...,groupN;][level=number;] Each component is optional, but they must appear the in order listed if applicable.
Matches:	[user=foo,bar,quux;group=manager,admin;level=100;], [group=nobody;level=24;]
Non-Matches:	[user=foo], [blahh]
Submitted By:	Michael Scovetta

Expression:	**^(\(?\+?[0-9]*\)?)?[0-9_\- \(\)]*$**

Matches:	[(+44)(0)20-12341234], [02012341234], [+44 (0) 1234-1234]	
Non-Matches:	[(44+)020-12341234], [12341234(+020)]	
Submitted By:	James Burton	

Expression: **\b(\w+)\s+\1\b**

Description: Uses backreferences and word boundaries to match repeated words seperated by whitespace without matching a word with the same ending as the next words beginning.

Matches: [Tell the the preacher], [some some], [hubba hubba]

Non-Matches: [once an annual report], [mandate dated submissions], [Hubba hubba]

Submitted By: Sean Carley

Expression:	**(^\+[0-9]{2}|^\+[0-9]{2}\(0\)|^\(\+[0-9]{2}\)\(0\)|^00[0-9]{2}|^0)([0-9]{9}$\|[0-9\-\s]{10}$)**	
Description:	Regular expression to evaluate dutch-style phone numbers. Possible example prefixes: +31, +31(0), (+31)(0), 0, 0031 followed by 9 numbers (which can contain a space or -).	
Matches:	[+31235256677], [+31(0)235256677], [023-5256677]	
Non-Matches:	[+3123525667788999], [3123525667788], [232-2566778]	
Submitted By:	Jon van Leuven	

Expression: **^[-+]?\d*\.?\d*$**

Description: Matches any floating point numer/numeric string, including optional sign character (+ or -). Also matches empty strings.

Matches: [123], [+3.14159], [-3.14159]

Non-Matches: [abc], [3.4.5], [$99.95]

Submitted By: Steven Smith

Expression:	**^\$?([1-9]{1}[0-9]{0,2}(\,[0-9]{3})*(\.[0-9]{0,2})?|[1-9]{1}[0-9]{0,}(\.[0-9]{0,2})?|0(\.[0-9]{0,2})?|(\.[0-9]{1,2})?)$**	
Description:	Many currency expresssions allow leading zeros, thus $01.40 passes thru them. This expression kills them, except for 0 in the one's column. Works with or without commas and/or dollar sign. Decimals not mandatory, unless no zero in ones column and decima ...	
Matches:	[$1,234.50], [$0.70], [.7]	
Non-Matches:	[$0,123.50], [$00.5]	
Submitted By:	Tom Persing	

Expression: **^[A-Z]{2}[0-9]{6}[A-DFM]{1}$**

Description: UK National Insurance Number (NINO) validation. (The following modifications have been made: Only A to D are permitted as the last

Matches: [AB123456D], [AB123456F], [AB123456M]

Non-Matches: [AB123456E], [ab123456d]

Non-Matches:	[AB123456E], [ab123456d]
Submitted By:	Peter Whyte

Expression:	**^[A-Z]{1,2}[1-9][0-9]?[A-Z]? [0-9][A-Z]{2,}|GIR 0AA$**
Description:	Matches UK postcodes according to the following rules 1. LN NLL eg N1 1AA 2. LLN NLL eg SW4 0QL 3. LNN NLL eg M23 4PJ 4. LLNN NLL eg WS14 0JT 5. LLNL NLL eg SW1N 4TB 6. LNL NLL eg W1C 8LQ. Modifications: 1) Doesn't allow leading zeros in first part (outw ...
Matches:	[G1 1AA], [GIR 0AA], [SW1 1ZZ]
Non-Matches:	[BT01 3RT], [G111 1AA]
Submitted By:	Peter Whyte

Expression:	**(http|ftp|https):\/\/[\w]+(.[\w]+)([\w\-\.,@?^=%&:/~\+#]*[\w\-\@?^=%&/~\+#])?**
Description:	This is the regular expression I use to add links in my email program. It also ignores those suppose-to-be commas/periods/colons at the end of the URL, like this sentence "check out http://www.yahoo.com/." (the period will be ignored) Note that ...
Matches:	[http://regxlib.com/Default.aspx], [http://electronics.cnet.com/electronics/0-6342366-8-8994967-1.html]
Non-Matches:	[www.yahoo.com]
Submitted By:	M H

Expression:	**^[0-9]{4}\s{0,1}[a-zA-Z]{2}$**
Description:	Regular expression to evaluate dutch zipcodes. This is an updated version of Roland Mensenkamp.
Matches:	[2034AK], [2034 AK], [2034 ak]
Non-Matches:	[2034 AK], [321321 AKSSAA]
Submitted By:	Jon van Leuven

Expression:	**((\d{2})|(\d))\/((\d{2})|(\d))\/((\d{4})|(\d{2}))**
Description:	This matches simple dates against 1 or 2 digits for the month, 1 or 2 digit for the day, and either 2 or 4 digits for the year
Matches:	[4/5/91], [04/5/1991], [4/05/89]
Non-Matches:	[4/5/1]
Submitted By:	Andrew Lader

Expression:	**(^|\s|\()(([1-9]{1}|([0][1-9]{1}|([1][012]){1}){1}[\/-]((2[0-9]{1}|(3[01]){1}|([01][1-**

Non-Matches:	[13/1/2001], [1-32-2001], [1-1-1801]
Submitted By:	David Good

Expression:	**^\d{3}\s?\d{3}$**
Description:	This can be used to match indian style pincodes / postal codes used by the indian postal departments which are 6 digits long and may have space after the 3rd digit
Matches:	[400 099], [400099], [400050]
Non-Matches:	[2345678], [12345], [asdf]
Submitted By:	anup kallingal

Expression:	**^\D?(\d{3})\D?\D?(\d{3})\D?(\d{4})$**
Description:	This RegEx requires a US phone number WITH area code. It is written to all users to enter whatever delimiters they want or no delimiters at all (i.e. 111-222-3333, or 111.222.3333, or (111) 222-3333, or 1112223333, etc...).
Matches:	[(111) 222-3333], [1112223333], [111-222-3333]
Non-Matches:	[11122223333], [11112223333], [11122233333]
Submitted By:	Laurence O'Donnell

Expression:	**^#?([a-f]|[A-F]|[0-9]){3}(([a-f]|[A-F]|[0-9]){3})?$**
Description:	Tests for valid HTML hexadecimal color codes. The # symbol is optional. And it will except either the 3 digit form for the 216 Web safe colors, or the full 6 digit form. I am use it on my site to allow users to customize the site's colors.
Matches:	[#00ccff], [#039], [ffffcc]
Non-Matches:	[blue], [0x000000], [#ff000]
Submitted By:	Chris Craft

Expression:	**^([0-9a-fA-F][0-9a-fA-F]:){5}([0-9a-fA-F][0-9a-fA-F])$**
Description:	This will grep for a valid MAC address , with colons seperating octets. It will ignore strings too short or long, or with invalid characters. It will accept mixed case hexadecimal. Use extended grep.
Matches:	[01:23:45:67:89:ab], [01:23:45:67:89:AB], [fE:dC:bA:98:76:54]
Non-Matches:	[01:23:45:67:89:ab:cd], [01:23:45:67:89:Az], [01:23:45:56:]
Submitted By:	Ted Rudyk

Expression:	**^(http|https|ftp)\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(:[a-zA-Z0-9]*)?/?([a-zA-Z0-9\-\._\?\,\'/\\\+&%\$#\=~])*$**
Description:	Modified URL RegExp that requires (http, https, ftp)://, A nice domain, and a decent file/folder string. Allows : after domain name, and these characters in the file/folder sring (letter, numbers, - . _ ? , ' / \ + & % $ # = ~). Blocks all other spec ...
Matches:	[http://www.blah.com/~joe], [ftp://ftp.blah.co.uk:2828/blah%20blah.gif], [https://blah.gov/blah-blah.as]
Non-Matches:	[www.blah.com], [http://www.blah"blah.com/I have spaces!], [ftp://blah_underscore/[nope]]
Submitted By:	Brandon Luhring

Expression:	**^(([0-2]\d|[3][0-1])\/([0]\d|[1][0-2])\/[2][0]\d{2})$\|^(([0-2]\d\|[3][0-1])\/([0]\d\|[1][0-2])\/[2][0]\d{2}\s([0-1]\d\|[2][0-3])\:[0-5]\d\:[0-5]\d)$**
Description:	Correct French DateTime(DD/MM/YYYY OR DD/MM/YYYY HH:MM:SS)
Matches:	[12/01/2002], [12/01/2002 12:32:10]
Non-Matches:	[32/12/2002], [12/13/2001], [12/02/06]
Submitted By:	Samir AZZA

Expression:	**^[0-9](\.[0-9]+)?$**
Description:	matches non-negative decimal floating points numbers less than 10
Matches:	[1.2345], [0.00001], [7]
Non-Matches:	[12.2], [1.10.1], [15.98]
Submitted By:	Snikwad Kcirtap

Expression:	**^(?:[mM]{1,3})?(?:(?:[cC][dDmM])|(?:[dD]?(?:[cC]{1,3})?))?[lL]?(([xX])(?:\2{1,2}|[lL]|[cC])?)?((([iI])(\5{1,2})|[vV]|[xX]|[lL])?)|([vV]?([iI]{1,3})?))?$**
Description:	This RE validates alpha characters that evaluate to Roman numerials, ranging from 1(I) - 3999(MMMCMXCIX). Not case sensitive.
Matches:	[III], [xiv], [MCMLXLIX]
Non-Matches:	[iiV], [MCCM], [XXXX]
Submitted By:	Michael Ash

Expression:	**^[-+]?[0-9]+[.]?[0-9]*([eE][-+]?[0-9]+)?$**
Description:	This matches floating point expression in a more rigorous way - accepts both exponent as well as non exponent notations.
Matches:	[123], [-123.35], [-123.35e-2]
Non-Matches:	[abc], [123.32e], [123.32.3]
Submitted By:	Srinivas Gummadi

Expression:	**^[a-zA-Z]+(([\'\,\.\-][a-zA-Z])?[a-zA-Z]*)*$**
Description:	Person's name (first, last, or both) in any letter case. Although not perfect, this expression will filter out many incorrect name formats (especially numerics and invalid special characters).
Matches:	[T.F. Johnson], [John O'Neil], [Mary-Kate Johnson]
Non-Matches:	[sam_johnson], [Joe--Bob Jones], [dfjsd0rd]
Submitted By:	Robert Dugre

Expression:	**^(20|21|22|23|[0-1]\d)[0-5]\d$**
Description:	This regular expression will match a 24 hour time with no separators.
Matches:	[1200], [1645], [2359]
Non-Matches:	[2400], [asbc], [12:45]

Submitted By:	Steven Smith
Expression:	**<[^>]*\n?.*=("|')?(.*\.jpg)("|')?.*\n?[^<]*>**
Description:	Match any image insert in a tag . simply replace the .jpg in the pattern whit a variable of content type ex:.swf,.js,.gif and loop the pattern to retrieve all tag whit the contenttype pass trought.... Very useful when you have people uploading html d ...
Matches:	[<td background="../img/img.jpg" >], [<img src=img.jpg >], [<img src='img.jpg']
Non-Matches:	[= img.jpg], [img.jpg]
Submitted By:	Marc-Antoine Latour
Expression:	**^(\d{5}-\d{4}|\d{5})$|^([a-zA-Z]\d[a-zA-Z] \d[a-zA-Z]\d)$**
Description:	This is a modification of the zip code regular expression submitted by Steven Smith (ssmith@aspalliance.com) It no longer matches 78754-12aA
Matches:	[78754], [78754-1234], [G3H 6A3]
Non-Matches:	[78754-12aA], [7875A], [g3h6a3]
Submitted By:	Khoi Pham
Expression:	**^([\w\-\.]+)@((\[([0-9]{1,3}\.){3}[0-9]{1,3}\])|(([\w\-]+\.)+)([a-zA-Z]{2,4}))$**
Description:	Expression 1 of 2 used to check email address syntax.
Matches:	[bob@somewhere.com], [bob.jones@[1.1.1.1]], [bob@a.b.c.d.info]
Non-Matches:	[bob@com], [bob.jones@some.where], [bob@1.1.1.123]
Submitted By:	David Lott
Expression:	**^(([-\w \.]+)|(""[-\w \.]+""))?<([\w\-\.]+)@((\[([0-9]{1,3}\.){3}[0-9]{1,3}\])|(([\w\-]+\.)+)([a-zA-Z]{2,4}))>$**
Description:	Expression 2 or 2 for matching email address syntax. This one matches the <angle bracket syntax>.
Matches:	[<ab@cd.ef>], [bob A. jones <ab@cd.ef>], [bob A. jones <ab@[1.1.1.111]>]
Non-Matches:	[ab@cd.ef], ["bob A. jones <ab@cd.ef>], [bob A. jones <ab@1.1.1.111>]
Submitted By:	David Lott
Expression:	**^[A-Za-z]{1,2}[0-9A-Za-z]{1,2}[ ]?[0-9]{0,1}[A-Za-z]{2}$**
Description:	This regular expression can be used to validate UK postcodes. Especially useful if want to provide a client side validation on a web site.
Matches:	[SW112LE], [SW11 2LE], [CR05LE]
Non-Matches:	[12CR0LE], [12CR 0LE], [SWLE05]
Submitted By:	Santosh Potadar
Expression:	**20\d{2}(-|\/)((0[1-9])|(1[0-2]))(-|\/)((0[1-9])|([1-2][0-9])|(3[0-1]))(T|\s)(([0-1][0-9])|(2[0-3])):([0-**
Description:	Sql date format tester.
Matches:	[2099-12-31T23:59:59], [2002/02/09 16:30:00], [2000-01-01T00:00:00]

Matches:	[2099-12-31T23:59:59], [2002/02/09 16:30:00], [2000-01-01T00:00:00]
Non-Matches:	[2000-13-31T00:00:00], [2002/02/33 24:00:00], [2000-01-01 60:00:00]
Submitted By:	Peter Pfeifer

Expression:	**^((?:4\d{3})|(?:5[1-5]\d{2})|(?:6011)|(?:3[68]\d{2})|(?:30[012345]\d))[-]?(\d{4})[-]?(\d{4})[-]?(\d{4}|3[4,7]\d{13})$**
Description:	This just a minor mod to Steven Smith's credit card re to accept spaces as separators, as well as return the four parts of the card. [Updated Oct-18-2002 to work with Diners Club/Carte Blanche (prefix must be 36, 38, or 300-305)]
Matches:	[6011567812345678], [6011 5678 1234 5678], [6011-5678-1234-5678]
Non-Matches:	[1234567890123456]
Submitted By:	Glenn Carr

Expression:	**^((((0[13578])|(1[02]))[\/]?(([0-2][0-9])|(3[01])))|(((0[469])|(11))[\/]?(([0-2][0-9])|(30)))|(02[\/]?[0-2][0-9]))[\/]?\d{4}$**
Description:	Date expressions that matches MM/DD/YYYY where MM and DD must be two digits and zero padded. Validates correctly for all months except February, which it assumes to always have 29 days. The "/" separator is optional.
Matches:	[01/01/2001], [02/29/2002], [12/31/2002]
Non-Matches:	[1/1/02], [02/30/2002], [1/25/2002]
Submitted By:	Steven Smith

Expression:	**^(?=[^&])(?:(?<scheme>[^:/?#]+):)?(?://(?<authority>[^/?#]*))?(?<path>[^?#]*)(?:\?(?<query>[^#]*))?(?:#(?<fragment>.*))?**
Description:	Use it for breaking-down a URI (URL, URN) reference into its main components: Scheme, Authority, Path, Query and Fragment. This is not a simple match regular expression. so it not works to verify a URI. It returns 1 matching group for each URI component ...
Matches:	[http://regexlib.com/REDetails.aspx?regexp_id=x#Details]
Non-Matches:	[&]
Submitted By:	Frederico Knabben

Expression:	**^[-+]?\d+(\.\d+)?$**
Description:	This matches any real number, with optional decimal point and numbers after the decimal, and optional positive (+) or negative (-) designation.
Matches:	[123], [-123.45], [+123.56]
Non-Matches:	[123x], [.123], [-123.]
Submitted By:	Steven Smith

Expression:	**^(\d{4}[-]){3}\d{4}|\d{16}$**

Matches:	[1234-1234-1234-1234], [1234 1234 1234 1234], [1234123412341234]	
Non-Matches:	[Visa], [1234], [123-1234-12345]	
Submitted By:	Steven Smith	

Expression:	**^((4\d{3})|(5[1-5]\d{2})|(6011))-?\d{4}-?\d{4}-?\d{4}|3[4,7]\d{13}$**	
Description:	Matches major credit cards including: Visa (length 16, prefix 4), Mastercard (length 16, prefix 51-55), Discover (length 16, prefix 6011), American Express (length 15, prefix 34 or 37). All 16 digit formats accept optional hyphens (-) between each grou ...	
Matches:	[6011-1111-1111-1111], [5423-1111-1111-1111], [341111111111111]	
Non-Matches:	[4111-111-111-111], [3411-1111-1111-111], [Visa]	
Submitted By:	Steven Smith	

Expression:	**^[A-Z0-9]{8}-[A-Z0-9]{4}-[A-Z0-9]{4}-[A-Z0-9]{4}-[A-Z0-9]{12}$**	
Description:	GUID Tester. It tests SQL Server GUIDs, which are alphanumeric characters grouped 8-4-4-4-12 (with the dashes). Make sure they don't have the brackets around them before you check them and have fun!	
Matches:	[4D28C5AD-6482-41CD-B84E-4573F384BB5C], [B1E1282C-A35C-4D5A-BF8B-7A3A51D9E388], [91036A4A-A0F4-43F0-8CD]	
Non-Matches:	[{B1E1282C-A35C-4D3A-BF8B-7A3A51D9E388}], [AAAAAAAAAAAAAAAA], [B;E1282C-A35C-4D3A-BF8B-7A3A51D9E38]	
Submitted By:	James Bray	

Expression:	**(^(4|5)\d{3}-?\d{4}-?\d{4}-?\d{4}|(4|5)\d{15})|(^(6011)-?\d{4}-?\d{4}-?\d{4}|(6011)-?\d{12})|(^((3\d{3}))-\d{6}-\d{5}|^((3\d{14}))**
Description:	This provides an expression to calidate the four major credit cards. It can be easily broken up to use for a specific type of card. It does not validate the number being a potential real number, only in the correct format.
Matches:	[4111-1234-1234-1234], [6011123412341234], [3711-123456-12345]
Non-Matches:	[1234567890123456], [4111-123-1234-1234], [412-1234-1234-1234]
Submitted By:	Chris Love

Expression:	**\[link="(?<link>((.|\n)*?))"\](?<text>((.|\n)*?))\[\/link\]**	
Description:	This can be used in conjunction with the replace method to provide pseudo-code support without having to enable HTML. The replacement string (in ASP.NET, use RegExp.Replace(SourceString, RegularExpressionPattern, ReplacementString) is <a href="${ ...	
Matches:	[[link="http://www.yahoo.com"]Yahoo[/link]]	
Non-Matches:	[[link]http://www.yahoo.com[/link]], [[link=http://www.yahoo.com]Yahoo[/link]]	
Submitted By:	Ryan S	

Expression:	**^[a-zA-Z0-9]+$**
Description:	Matches any alphanumeric string (no spaces).
Matches:	[10a], [ABC], [A3fg]
Non-Matches:	[45.3], [this or that], [$23]
Submitted By:	Steven Smith

Expression:	**((\(\d{3}\) ?)|(\d{3}-))?\d{3}-\d{4}**
Description:	US Phone Number -- doesn't check to see if first digit is legal (not a 0 or 1).
Matches:	[(123) 456-7890], [123-456-7890]
Non-Matches:	[1234567890]
Submitted By:	Steven Smith

Expression:	**^[a-zA-Z]\w{3,14}$**
Description:	The password's first character must be a letter, it must contain at least 4 characters and no more than 15 characters and no characters other than letters, numbers and the underscore may be used
Matches:	[abcd], [aBc45DSD_sdf], [password]
Non-Matches:	[afv], [1234], [reallylongpassword]
Submitted By:	Steven Smith

Expression:	**^0[23489]{1}(\-)?[^0\D]{1}\d{6}$**
Description:	Regular Expression that validate a phone number inside israel.
Matches:	[03-6106666], [036106666], [02-5523344]
Non-Matches:	[00-6106666], [03-0106666], [02-55812346]
Submitted By:	Al Val

Expression:	**^0(5[012345678]|6[47]){1}(\-)?[^0\D]{1}\d{5}$**
Description:	Regular Expression that validate Cellular phone in israel.
Matches:	[050-346634], [058633633], [064-228226]
Non-Matches:	[059-336622], [064-022663], [0545454545]
Submitted By:	Al Val

Expression:	**^([A-Z]{1,2}[0-9]{1,2}|[A-Z]{3}|[A-Z]{1,2}[0-9][A-Z])(|-)[0-9][A-Z]{2}**
Description:	Checks whether the string specified is in the same format as the UK postcode format defined on: http://www.magma.ca/~djcl/postcd.txt It allows: A = Letter N = Number AN NAA, ANA NAA, ANN NAA, AAN NAA, AAA NAA (rare), AANN NAA, AANA NAA It gives the ...
Matches:	[AA11 1AA], [AA1A 1AA], [A11-1AA]
Non-Matches:	[111 AAA], [1AAA 1AA], [A1AA 1AA]
Submitted By:	Henk de Vries -op regexlib.com-

Expression:	**@{2}((\S)+)@{2}**
Description:	This will match results in a template situation. For example: template reads Dear @@Name@@, ... would become Dear John, If you dont want to use the @@ change the @ to what ever characters you want.
Matches:	[@@test@@], [@@name@@], [@@2342@@]
Non-Matches:	[@test@], [@@na me@@], [@@ name@@]

Submitted By:	john hamman
Expression:	**([0-1][0-9]|2[0-3]):[0-5][0-9]**
Description:	Validate an hour entry to be between 00:00 and 23:59
Matches:	[00:00], [13:59], [23:59]
Non-Matches:	[24:00], [23:60]
Submitted By:	Steve Charland
Expression:	**^[+-]?([0-9]*\.?[0-9]+|[0-9]+\.?[0-9]*)([eE][+-]?[0-9]+)?$**
Description:	A regular expression that matches numbers. Integers or decimal numbers with or without the exponential form.
Matches:	[23], [-17.e23], [+.23e+2]
Non-Matches:	[+.e2], [23.17.5], [10e2.0]
Submitted By:	Erik Pettersson
Expression:	**^([1-zA-Z0-1@.\s]{1,255})$**
Description:	A general string validation to insure no malicious code is being passed through user input. General enough too allow email address, names, address, passwords, so on. Disallows `',*&$<> or other characters that could cause issues.
Matches:	[email@email.com], [My Name], [asdf12df]
Non-Matches:	[',*&$<>], [1001' string]
Submitted By:	Paul Miller
Expression:	**^((0[1-9])|(1[0-2]))\/(\d{4})$**
Description:	This regular expressions matches dates of the form MM/YYYY where MM can be 01 to 12 and YYYY is always 4 digits long.
Matches:	[12/2002], [11/1900], [02/1977]
Non-Matches:	[1/1977], [00/000], [15/2002]
Submitted By:	Jony Ferreira
Expression:	**^\(\d{1,2}(\s\d{1,2}){1,2}\)\s(\d{1,2}(\s\d{1,2}){1,2})((-(\d{1,4})){0,1})$**
Description:	Meets german norm-standard: DIN 5008: 1996-05 for telephone numbers
Matches:	[(0 34 56) 34 56 67], [(03 45) 5 67 67], [(0 45) 2 33 45-45]
Non-Matches:	[(2345) 34 34], [(0 56) 456 456], [(3 45) 2 34-45678]
Submitted By:	Gerald Martin
Expression:	**(?:\d|I{1,3})?\s?\w{2,}\.?\s*\d{1,}\:\d{1,}-?,?\d{0,2}(?:,\d{0,2}){0,2}**
Description:	This RA validates standard Bible verse notation.
Matches:	[Genesis 3:3-4,6], [II Sam 2:11,2], [2 Tim 3:16]
Non-Matches:	[Genesis chap 3, verse 3], [2nd Samuel 2]
Submitted By:	Scott Kahler

Expression:	**(\[[Ii][Mm][Gg]\])(\S+?)(\[\/[Ii][Mm][Gg]\])**
Description:	easy when you want to allow your users to post images, but in a controlled way. I used it like this (in php): $text = preg_replace("/(\[IMG\])(\S+?)(\[\/IMG\])/is", "<a href=\"\\2\" target=\"_blank\"><IMG SR ...
Matches:	[[IMG]http://bleh.jpg[/IMG]], [[ImG]bleh[/imG]], [[img]ftp://login:pass@bleh.gif[/img]]
Non-Matches:	[<img src="bleh.jpg">]
Submitted By:	marnik vander elst

Expression:	**^([0-9]{1,2})[./-]+([0-9]{1,2})[./-]+([0-9]{2}|[0-9]{4})$**
Description:	Validate date formats: dd/mm/yyyy or d/m/yy or d.m.yyyy with separators: . - / (rizzipereira.com.br)
Matches:	[10/03/1979], [1-1-02], [01.1.2003]
Non-Matches:	[10/03/197], [01-02-003], [01 02 03]
Submitted By:	Diego dos Santos

Expression:	**^(?(^00000(|-0000))|(\d{5}(|-\d{4})))$**
Description:	US zip code expression that disallows 00000 or 00000-0000 for either ZIP or ZIP+4. Great for web site validation.
Matches:	[12345], [12345-6789]
Non-Matches:	[00000], [00000-0000], [a4650-465s]
Submitted By:	Paul Duncan

Expression:	**^([0-1]([\s-./\\])?)?(\(?[2-9]\d{2}\)?|[2-9]\d{3})([\s-./\\])?(\d{3}([\s-./\\])?\d{4}|[a-zA-Z0-9]{7})$**
Description:	It checks for Valid US Phone numbers. Accepts ".",Space,\,/,- as delim.
Matches:	[1.222.333.1234], [1-223-123-1232], [12223334444]
Non-Matches:	[1.1.123123.123], [12-1322-112-31], [11231321131]
Submitted By:	Amit Deshpande

Expression:	**^((0?[1-9])|((1|2)[0-9])|30|31)$**
Description:	matches any day of month 0?1-31
Matches:	[01], [12], [31]
Non-Matches:	[123], [32], [abc]
Submitted By:	Serafeim Zanikolas

Expression:	**^(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{4,8}$**
Description:	Password matching expression. Password must be at least 4 characters, no more than 8 characters, and must include at least one upper case letter, one lower case letter, and one numeric digit.
Matches:	[asD1], [asDF1234], [ASPgo123]
Non-Matches:	[asdf], [1234], [ASDF12345]

Submitted By:	Steven Smith
Expression:	**^(([0]?[1-9]|1[0-2])(:)([0-5][0-9]))$**
Description:	This time validation expression accepts an hour between 1 and 9 (with optional leading zero) and minutes between 01 and 59. This is primarily for use with an AM/PM drop down list or radio button.
Matches:	[09:00], [9:00], [11:35]
Non-Matches:	[13:00], [9.00], [6:60]
Submitted By:	Ken Madden
Expression:	**^([1-9]|[1-9]\d|1\d{2}|2[0-4]\d|25[0-5])$**
Description:	This validates a number between 1 and 255. Could be modified to IP, or just to verify a number in a range.
Matches:	[1], [108], [255]
Non-Matches:	[01], [256]
Submitted By:	K Thompson
Expression:	**^((((0[13578])|([13578])|(1[02]))[\/](([1-9])|([0-2][0-9])|(3[01])))|(((0[469])|([469])|(11))[\/](([1-9])|([0-2][0-9])|(30)))|((2|02)[\/](([1-9])|([0-2][0-9]))))[\/]\d{4}$\|^\d{4}$**
Description:	The following validates dates with and without leading zeros in the following formats: MM/DD/YYYY and it also takes YYYY (this can easily be removed). All months are validated for the correct number of days for that particular month except for February ...
Matches:	[01/01/2001], [1/01/2001], [2002]
Non-Matches:	[2/30/2002], [13/23/2002], [12345]
Submitted By:	Matt Adams
Expression:	**^[A-Za-z]{2}[0-9]{6}[A-Za-z]{1}$**
Description:	UK National Insurance Number validation. Especially useful to validate through clientside/server side script on a website.
Matches:	[SP939393H], [PX123456D], [SW355667G]
Non-Matches:	[12SP9393H], [S3P93930D], [11223344SP00ddSS]
Submitted By:	Santosh Potadar
Expression:	**(^0[78][2347][0-9]{7})**
Description:	checks for valid South African cellular numbers
Matches:	[0834128458], [0749526308]
Non-Matches:	[0861212308], [0892549851]
Submitted By:	Zahir Jacobs
Expression:	**^([A-HJ-TP-Z]{1}\d{4}[A-Z]{3}|[a-z]{1}\d{4}[a-hj-tp-z]{3})$**
Description:	Codigos Postales Argentinos (CPA) This expression defines the new zip code format for Argentina.
Matches:	[C1406HHA], [A4126AAB], [c1406hha]
Non-Matches:	[c1406HHA], [4126], [C1406hha]

Non-Matches:	[c1406HHA], [4126], [C1406hha]
Submitted By:	Gabriel Garcia

Expression:	**^(((25[0-5]|2[0-4][0-9]|19[0-1]|19[3-9]|18[0-9]|17[0-1]|17[3-9]|1[0-6][0-9]|1[1-9]|[2-9][0-9]|[0-9])\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9]))|(192\.(25[0-5]|2[0-4][0-9]|16[0-7]|169|1[0-5][0-9]|1[7-9][0-9]|[1-9][0-9]|[0-9]))|(172\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|1[0-5]|3[2-9]|[4-9][0-9]|[0-9])))\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])\.(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9][0-9]|[0-9])$**
Description:	[Routable IP Addresses] Works great with ASP.NET RegularExpressionValidator. This expression excludes all private ip address blocks specified in RFC 1918 (listed below). If you want to verify an IP address, but you do not want private, non-routabl ...
Matches:	[66.129.71.120], [207.46.230.218], [64.58.76.225]
Non-Matches:	[10.0.5.4], [192.168.0.1], [my ip address]
Submitted By:	Todd Krabach

Expression:	**^([\w\d\-\.]+)@{1}(([\w\d\-]{1,67})|([\w\d\-]+\.[\w\d\-]{1,67}))\.(([a-zA-Z\d]{2,4})(\.[a-zA-Z\d]{2})?)$**
Description:	This pattern allows standard e-mail addresses (e.g. user@domain.com), sub domains (e.g. user@foo.domain.com), the new two- and four-letter domains (e.g. user@domain.tv and user@domain.name) and country codes (e.g. user@foo.com.us). Also, this patter foll ...
Matches:	[foo@foo.com], [foo@foo-foo.com.au], [foo@foo.foo.info]
Non-Matches:	[foo@.com], [foo@foo..com], [foo@me@.com]
Submitted By:	Laurence O'Donnell

Expression:	**/*[\d\D]*?*/**
Description:	If you need to extract or remove any /* */ sytle comments from any Java, JavaScript, C, C++, CSS, etc code you have this regular expression can help.
Matches:	[/* my comment */], [/* my multiline comment */], [/* my nested comment */]
Non-Matches:	[*/ anything here /*], [anything between 2 seperate comments], [* *\]
Submitted By:	Chris Craft

Expression:	**((0?[13578]|10|12)(-|\/)((0[0-9])|([12])([0-9]?)|(3[01]?))(-|\/)((\d{4})|(\d{2}))|(0?[2469]|11)(-|\/)((0[0-9])|([12])([0-9]?)|(3[0]?))(-|\/)((\d{4}|\d{2}))**
Description:	Regex used in .NET to validate a date. Matches the following formats mm/dd/yy, mm/dd/yyyy, mm-dd-yy, mm-dd-yyyy This covers days with 30 or 31 days but does not handle February, it is allowed 30 days.
Matches:	[1/31/2002], [04-30-02], [12-01/2002]
Non-Matches:	[2/31/2002], [13/0/02], [Jan 1, 2001]
Submitted By:	Sean McAnally

Expression:	**^(([^<>;()[\]\\.,;:@"]+(\.[^<>()[\]\\.,;:@"]+)*)|(".+"))@((([a-z]([-a-z0-9]*[a-z0-9])?)|(#[0-**

	9]+)|(\[(((([01]?[0-9]{0,2})|(2(([0-4][0-9])|(5[0-5])))\.){3}(([01]?[0-9]{0,2})|(2(([0-4][0-9])|(5[0-5])))\]))\.)*(([a-z]([-a-z0-9]*[a-z0-9])?)|(#[0-9]+)|(\[(((([01]?[0-9]{0,2})|(2(([0-4][0-9])|(5[0-5])))\.){3}(([01]?[0-9]{0,2})|(2(([0-4][0-9])|(5[0-5])))\]))$
Description:	This regular expression matches email addresses that match the format specified by RFC 821 (see the BNF notation at http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc0821.html#page-30)
Matches:	[blah@[10.0.0.1]], [a@b.c]
Non-Matches:	[non@match@.]
Submitted By:	Jason Howat

Expression:	**^\d{9}[\d|X]$**
Description:	A very simple ISBN validation expression - it just checks for a 10 digit number where the last digit could also be a capital 'X'. Complete specs for ISBN available here: http://www.isbn.org/standards/home/isbn/international/html/usm4.htm. An enhanceme ...
Matches:	[1234123412], [123412341X]
Non-Matches:	[not an isbn]
Submitted By:	Steven Smith

Expression:	**^(([1-9])|(0[1-9])|(1[0-2]))\/(([0-9])|([0-2][0-9])|(3[0-1]))\/(([0-9][0-9])|([1-2][0,9][0-9][0-9]))$**
Description:	Dates day: d or dd, <= 31, month: m or mm, <= 12, year: yy or yyyy >= 1900, <= 2099
Matches:	[01/01/2001], [1/1/1999], [10/20/2080]
Non-Matches:	[13/01/2001], [1/1/1800], [10/32/2080]
Submitted By:	Scott Thompson

Expression:	**^\d*\.?((25)|(50)|(5)|(75)|(0)|(00))?$**
Description:	This is a pattern to search and verify that a decimal number ends with a 25, 50, 75, 0 or 00. It does match for a nothing after decimal also but I guess thats ok !!
Matches:	[0.25], [.75], [123.50]
Non-Matches:	[.77], [1.435]
Submitted By:	narsi v

Expression:	**^(s-|S-){0,1}[0-9]{3}\s?[0-9]{2}$**
Description:	Validates swedish zipcodes (postnr) with or without space between groups. With leading s- or not. Can be disconnected by removing "(s-|S-){0,1}".
Matches:	[12345], [932 68], [S-621 46]
Non-Matches:	[5367], [425611], [31 545]
Submitted By:	Tommy Ullberg

Expression:	**^\d{5}(-\d{4})?$**
Description:	Matches standard 5 digit US Zip Codes, or the US ZIP + 4 Standard.

Matches:	[48222], [48222-1746]
Non-Matches:	[4632], [Blake], [37333-32]
Submitted By:	Blake Facey

Expression:	**^(?!^(PRN|AUX|CLOCK\$|NUL|CON|COM\d|LPT\d|\..*)(\..+)?$)[^\x00-\x1f\\?*:\";\|/]+$**
Description:	Checks for a valid windows file name (Must be used with the case-insensitive option Checks that the file has at lease one char, does not contain any invalid characters and does not have a reserved word as a file name. "If you see a " ...
Matches:	[test.txt], [test.jpg.txt], [a&b c.bmp]
Non-Matches:	[CON], [.pdf], [test:2.pdf]
Submitted By:	Andre Van Der Merwe

Expression:	**^(\d{1,3}'(\d{3}')*\d{3}(\.\d{1,3})?|\d{1,3}(\.\d{3})?)$**
Description:	This regex match numeric data in the following format: thousands are separated by (') apostrophe, decimal places are separated by dot (.) Maximum three decimal places are not required. It's easy to change to other separators as well.
Matches:	[1'235.140], [1'222'333.120], [456]
Non-Matches:	[1234.500], [78'45.123], [123,0012]
Submitted By:	Dalibor Kalna

Expression:	**^[a-zA-Z][0-9][a-zA-Z]\s?[0-9][a-zA-Z][0-9]$**
Description:	Match Canadia Zip Code. You can have a space in the middle Like T2P 3C7, or no space like T2P3C7
Matches:	[T2p 3c7], [T3P3c7], [T2P 3C7]
Non-Matches:	[123456], [3C7T2P], [11T21RWW]
Submitted By:	Qing Jiang

Expression:	**^\$[0-9]+(\.[0-9][0-9])?$**
Description:	Validates a dollar amount including a dollar sign and 2 decmals. The decimal and cents are optional.
Matches:	[$1.50], [$49], [$0.50]
Non-Matches:	[1.5], [$1.333], [this $5.12 fails]
Submitted By:	Bob Levittan

Expression:	**\b((25[0-5]|2[0-4]\d|[01]\d\d|\d?\d)\.){3}(25[0-5]|2[0-4]\d|[01]\d\d|\d?\d)\b**
Description:	Most Concise RegExp for matching Decimal IPs. If nothing else, it'll make your code easier to read. (And I know that \d?\d is \d{1,2} but that's 2 extra characters.)
Matches:	[217.6.9.89], [0.0.0.0], [255.255.255.255]
Non-Matches:	[256.0.0.0], [0978.3.3.3], [65.4t.54.3]
Submitted By:	Sean Schricker

Expression:	**((mailto\:|(news|(ht|f)tp(s?))\://){1}\S+)**
Description:	Regular Expression matches any internet URLs. Used with the replace method it comes in very handy.
Matches:	[http://www.aspemporium.com], [mailto:dominionx@hotmail.com], [ftp://ftp.test.com]
Non-Matches:	[www.aspemporium.com], [dominionx@hotmail.com], [bloggs]
Submitted By:	Justin Saunders
Expression:	**\(([0-9]{2}|0{1}((x|[0-9]){2}[0-9]{2}))\)\s*[0-9]{3,4}[-]*[0-9]{4}**
Description:	Match diferent styles for brazilian Phone number code. Only DDD (12), complete DDD (012), complete DDD + Telephony Company (0xx12) plus 3 or 4 digits (city code) plus 4 digits (phone number).
Matches:	[(12) 123 1234], [(01512) 123 1234], [(0xx12) 1234 1234]
Non-Matches:	[12 123 1234], [(012) 123/1234], [(012) 123 12345]
Submitted By:	Samuel Mota
Expression:	**^\w+[\w-\.]*\@\w+((-\w+)|(\w*))\.[a-z]{2,3}$**
Description:	Email validation. With this short expression you can validate for proper email format. It's short and accurate.
Matches:	[bob-smith@foo.com], [bob.smith@foo.com], [bob_smith@foo.com]
Non-Matches:	[-smith@foo.com], [.smith@foo.com], [smith@foo_com]
Submitted By:	Eric Lebetsamer
Expression:	**^(?=.*\d).{4,8}$**
Description:	Password expression. Password must be between 4 and 8 digits long and include at least one numeric digit.
Matches:	[1234], [asdf1234], [asp123]
Non-Matches:	[asdf], [asdf12345], [password]
Submitted By:	Steven Smith
Expression:	**[^A-Za-z0-9_@\.]|@{2,}|\.{5,}**
Description:	Used as a username validation script requires: 1. Allows All Alphanumeric characters & underscore 2. Allows One "@" character 3. Allows Five "." periods 4. Rejects spaces
Matches:	[user name], [user#name], [.....]
Non-Matches:	[User_Name1], [username@foo.com], [user.name@mail.foo.com]
Submitted By:	Theo Tillotson
Expression:	**^100$\|^[0-9]{1,2}$|^[0-9]{1,2}\,[0-9]{1,3}$**
Description:	Percentage with 3 number after comma.
Matches:	[12,654], [1,987]
Non-Matches:	[128,2], [12,]
Submitted By:	Samir AZZA
Expression:	**^(http|https|ftp)\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(:[a-zA-Z0-9]*)?/?([a-zA-Z0-9\-**

	\._\?\,\'/\\\+&%\$#\=~])*[^\.\,\)\(\s]$
Description:	This Regex (can be used e.g. in PHP with eregi) will match any valid URL. Unlike the other exapmles here, it will NOT match a valid URL ending with a dot or bracket. This is important if you use this regex to find and "activate" Links in an Text
Matches:	[https://www.restrictd.com/~myhome/]
Non-Matches:	[http://www.krumedia.com.], [(http://www.krumedia.com)], [http://www.krumedia.com,]
Submitted By:	Michael Krutwig

Expression:	**(\d{1,3},(\d{3},)*\d{3}(\.\d{1,3})?|\d{1,3}(\.\d{3})?)$**
Description:	Dollar Amount
Matches:	[2&651.50], [987.895]
Non-Matches:	[25$%787*]
Submitted By:	Snikwad Kcirtap

Expression:	**\$[0-9]?[0-9]?[0-9]?((\,[0-9][0-9][0-9])*)?(\.[0-9][0-9]?)?$**
Description:	
Matches:	[$1,456,983.00], [$1,700.07], [$68,944.23]
Non-Matches:	[$20,86.93], [$1098.84], [$150.]
Submitted By:	Snikwad Kcirtap

Expression:	**\$[0-9]?[0-9]?[0-9]?((\,[0-9][0-9][0-9])*)?(\.[0-9][0-9])?$**
Description:	
Matches:	[$28,009,987.88], [$23,099.05], [$.88]
Non-Matches:	[$234,5.99]
Submitted By:	yahnique barrett

Expression:	**^((((31\/(0?[13578]|1[02]))|((29|30)\/(0?[1,3-9]|1[0-2])))\/(1[6-9]|[2-9]\d)?\d{2})|(29\/0?2\/(((1[6-9]|[2-9]\d)?(0[48]|[2468][048]|[13579][26])|((16|[2468][048]|[3579][26])00))))|(0?[1-9]|1\d|2[0-8])\/((0?[1-9])|(1[0-2]))\/((1[6-9]|[2-9]\d)?\d{2})) (20|21|22|23|[0-1]?\d):[0-5]?\d:[0-5]?\d$**
Description:	this expression validates a date-time field in European d/m/y h:m:s format. It is a european variation of Michael Ash's date-validation expression in this library. The days are validated for the given month and year. Leap years are validated for all 4 d ...
Matches:	[29/02/2004 20:15:27], [29/2/04 8:9:5], [31/3/2004 9:20:17]
Non-Matches:	[29/02/2003 20:15:15], [2/29/04 20:15:15], [31/3/4 9:20:17]
Submitted By:	martin devos

Expression:	**^([a-zA-Z0-9_\-\.]+)@([a-zA-Z0-9_\-\.]+)\.([a-zA-Z]{2,5})$**
Description:	Easy expression that checks for valid email addresses.

Description:	Easy expression that checks for valid email addresses.
Matches:	[somthing@someserver.com], [firstname.lastname@mailserver.domain.com], [username-something@some-server.]
Non-Matches:	[username@someserver.domain.c], [somename@server.domain-com], [someone@something.se_eo]
Submitted By:	Zrekam makerZ

Expression:	**(^([0-9]|[0-1][0-9]|[2][0-3]):([0-5][0-9])(\s{0,1})(AM|PM|am|pm|aM|Am|pM|Pm{2,2})$)\|(^([0-9]\|[1][0-9]\|[2][0-3])(\s{0,1})(AM\|PM\|am\|pm\|aM\|Am\|pM\|Pm{2,2})$)**
Description:	Validate "Time" Data to Work with SQL Server This is a fix (I hope) for a problem with the original expression. It originally allowed any combination of am or pm. For example: ma, aa, mm, mp, etc.
Matches:	[8am], [8 am], [8:00 am]
Non-Matches:	[8a], [8 a], [8:00 a]
Submitted By:	Greg Burns

Expression:	**^([0-9]{2})?(\([0-9]{2})\)([0-9]{3}|[0-9]{4})-[0-9]{4}$**
Description:	A simple expression to brazilian phone number code, with international code. Simple DDI without "+" 99 plus simple DDD (99) plus simple local phone number 3 or 4 digits plus "-" plus 4 digits.
Matches:	[55(21)123-4567], [(11)1234-5678], [55(71)4562-2234]
Non-Matches:	[3434-3432], [4(23)232-3232], [55(2)232-232]
Submitted By:	Rafael Miranda

Expression:	**^((([0]?[1-9]|1[0-2])(:|\.)[0-5][0-9]((:|\.)[0-5][0-9])?( )?(AM|am|aM|Am|PM|pm|pM|Pm))|(([0]?[0-9]|1[0-9]|2[0-3])(:|\.)[0-5][0-9]((:|\.)[0-5][0-9])?))$**
Description:	Matches times seperated by either : or . will match a 24 hour time, or a 12 hour time with AM or PM specified. Allows 0-59 minutes, and 0-59 seconds. Seconds are not required.
Matches:	[1:01 AM], [23:52:01], [03.24.36 AM]
Non-Matches:	[19:31 AM], [9:9 PM], [25:60:61]
Submitted By:	Steve Valaitis

Expression:	**^\d{0,2}(\.\d{1,2})?$**
Description:	This regular expression validates that the data entered is a number with a maximum of two integers and two decimals and a minimum of one integer or one decimal.
Matches:	[99.99], [99], [.99]
Non-Matches:	[999.999], [999], [.999]
Submitted By:	Jaime Borges

Expression:	**^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?!.*\s).{4,8}$**
Description:	Password expresion that requires one lower case letter, one upper case letter, one digit, 6-13 length, and no spaces. This is merely an
Matches:	[1agdA*$#], [1agdA*$#], [1agdA*$#]

Matches:	[1agdA*$#], [1agdA*$#], [1agdA*$#]
Non-Matches:	[wyrn%@*&$# f], [mbndkfh782], [BNfhjdhfjd&*)%#$)]
Submitted By:	Jeremy Samuel

Expression:	**^([a-zA-Z0-9][-a-zA-Z0-9]*[a-zA-Z0-9]\.)+([a-zA-Z0-9]{3,5})$**
Description:	Host/Domain name validation for perl. Should be combined with a check for length <= 63 characters and that $2 is in a list of top-level domains.
Matches:	[freshmeat.net], [123.com], [TempLate-toolkKt.orG]
Non-Matches:	[-dog.com], [?boy.net], [this.domain]
Submitted By:	Nathan Pitts

Expression:	**^[^']*$**
Description:	This one matches all strings that do not contain the single quotation mark (').
Matches:	[asljas], [%/&89uhuhadjkh], ["hi there!"]
Non-Matches:	['hi there!'], [It's 9 o'clock], [""]
Submitted By:	Peter Friese

Expression:	**(^\(\)$\|^\((\([0-9]+,(\((\([0-9]+,[0-9]+,[0-9]+\),)*(\([0-9]+,[0-9]+,[0-9]+\)){1}\))+\),)*(\([0-9]+,(\((\([0-9]+,[0-9]+,[0-9]+\),)*(\([0-9]+,[0-9]+,[0-9]+\)){1}\))+\)){1}\))$**
Description:	This checks for the specific syntax ((A,((b,c,d),(e,f,g))), ..). No limit on number of occurances.
Matches:	[((24,((1,2,3),(3,4,5))))], [((1,((2,3,4),(4,5,6),(96,34,26))),(12,((1,3,4),(4,5,6),(7,8,9))))], [()]
Non-Matches:	[(24,((1,2,3),(3,4,5)))], [( )], [((23,(12,3,4),(4,5,6)))]
Submitted By:	Rambabu Tummala

Expression:	**^[a-zA-Z0-9\s.\-_']+$**
Description:	Alphanumeric, hyphen apostrophe, comma dash spaces
Matches:	[dony d'gsa]
Non-Matches:	[^[a-zA-Z0-9\s.\-_']+$]
Submitted By:	Sonal Bordia

Expression:	**^[_a-zA-Z0-9-]+(\.[_a-zA-Z0-9-]+)*@[a-zA-Z0-9-]+(\.[a-zA-Z0-9-]+)*\.(([0-9]{1,3})|([a-zA-Z]{2,3})|(aero|coop|info|museum|name))$**
Description:	Matches e-mail addresses, including some of the newer top-level-domain extensions, such as info, museum, name, etc. Also allows for emails tied directly to IP addresses.
Matches:	[example@example.com], [foo@bar.info], [blah@127.0.0.1]
Non-Matches:	[broken@@example.com], [foo@bar.infp], [blah@.nospam.biz]
Submitted By:	David Huyck

Expression:	**^\d{5}(-\d{3})?$**
Description:	Matches standard 5 digit Brazilian Postal Codes (CEP), or the CEP + 3 digits (distribution identifiers - suffix). For more info refer to: http://www.correios.com.br/servicos/cep/Estrutura_CEP.cfm (in portuguese).
Matches:	[13165-000], [38175-000], [81470-276]
Non-Matches:	[13165-00], [38175-abc], [81470-2763]
Submitted By:	Carlos Nascimento

Expression:	**^\$(\d{1,3}(\,\d{3})*|(\d+))(\.\d{2})?$**
Description:	This re matches US currency format with lead dollar sign. Dollar value must have at least one digit and may or may not be comma separated. Cents value is optional.
Matches:	[$0.84], [$123458], [$1,234,567.89]
Non-Matches:	[$12,3456.01], [12345], [$1.234]
Submitted By:	Michael Ash

Expression:	**([A-Z]:\\[^/:*\?<>|]+\.\w{2,6})|(\\{2}[^/:*\?<>|]+\.\w{2,6})**
Description:	This regular expression pattern can be used to check the validity of paths for file upload controls. The uploaded file can be either stored locally or accessible through UNC. It cannot contain illegal characters for the windows OS - that may be supported ...
Matches:	[C:\temp\this allows spaces\web.config], [\\Andromeda\share\file name.123]
Non-Matches:	[tz:\temp\ fi*le?na:m<e>.doc], [\\Andromeda\share\filename.a]
Submitted By:	Alban Schmid

Expression:	**(^([0-9]|[0-1][0-9]|[2][0-3]):([0-5][0-9])$)\|(^([0-9]\|[1][0-9]\|[2][0-3])$)**
Description:	Matches a string if it is a valid time in the format of HH:MM / H:MM / HH / H
Matches:	[10:35], [9:20], [23]
Non-Matches:	[24:00], [20 PM], [20:15 PM]
Submitted By:	Nacho Varas

Expression:	**^\$?([0-9]{1,3},([0-9]{3},)*[0-9]{3}|[0-9]+)(\.[0-9][0-9])?$**
Description:	Matches US currency input with or without commas. This provides a fix for the currency regular expression posted at http://regxlib.com/REDetails.aspx?regexp_id=70 by escaping the . (period) to ensure that no other characters may be used in it's place.
Matches:	[$3,023,123.34], [9,876,453], [123456.78]
Non-Matches:	[4,33,234.34], [$1.234], [abc]
Submitted By:	Al Kahler

Expression:	**^\$?\d+(\.(\d{2}))?$**
Description:	To evaluate an amount with or without a dollar sign where the cents are optional.

Matches:	[$2.43], [2.02], [$2112]
Non-Matches:	[2.1], [$.14], [$2,222.12]
Submitted By:	Daniel Repaci

Expression:	**((0[1-9])|(1[02]))/\d{2}**
Description:	Fromat check for MM/YY, checks month is 1-12 and any 2 digit year.
Matches:	[01/00], [12/99]
Non-Matches:	[13/00], [12/AS]
Submitted By:	Andrew Balaschak

Expression:	**("[^"]*")|('[^\r]*)(\r\n)?**
Description:	Will match a VBScript string and/or comment Ex: ' userinfo strUsername = "tomsve" iAge = 20 ' temp strPassword = "halloj" ...Would result in the following matches: ' userinfo "tomsve" ' temp "halloj& ...
Matches:	["my string"], ["a string with ' in it"], [' comment]
Non-Matches:	[asd "]
Submitted By:	Tom Svensson

Expression:	**^[A-Za-z0-9]{8}-[A-Za-z0-9]{4}-[A-Za-z0-9]{4}-[A-Za-z0-9]{4}-[A-Za-z0-9]{12}$**
Description:	GUID Tester. This is a modification from the regular expression submitted by James Bray (james@jamesbray.com). It allows the use of mixed upper and lowercase letters in the GUID string.
Matches:	[BFDB4D31-3E35-4DAB-AFCA-5E6E5C8F61EA], [BFDB4d31-3e35-4dab-afca-5e6e5c8f61ea]
Non-Matches:	[qqqBFDB4D31-3E35-4DAB-AFCA-5E6E5C8F61EA], [BFDB4D31-3E-4DAB-AFCA-5E6E5C8F61EA], [BFDB4D31-3E35-4DAB-AF]
Submitted By:	V. Lorz

Expression:	**^\d{2}(\x2e)(\d{3})(-\d{3})?$**
Description:	Other expression to standard 5 digit Brazilian Postal Codes (CEP), or the CEP + 3 digits (distribution identifiers - suffix). The diference of the original one, is that the "." is mandatory.
Matches:	[12.345-678], [23.345-123], [99.999]
Non-Matches:	[41222-222], [3.444-233], [43.324444]
Submitted By:	Rafael Miranda

Expression:	**^(([a-zA-Z]:)|(\\{2}\w+)\$?)(\\(\w[\w]*))+\.(txt|TXT)$**
Description:	This RE validates a path/file of type txt (text file) This RE can be used as a filter on certain file types, while insuring the entire string is a fully qualified path and file. The filter value can be changed or added to as you need
Matches:	[c:\file.txt], [c:\folder\sub folder\file.txt], [\\network\folder\file.txt]
Non-Matches:	[C:], [C:\file.xls], [folder.txt]
Submitted By:	Michael Ash

Expression:	**^[a-zA-Z0-9]+([a-zA-Z0-9\-\.]+)?\.(com|org|net|mil|edu|COM|ORG|NET|MIL|EDU)$**

Description:	Checks domain names. This is an attempt to deal with some of the issues of the other reg ex in not handling leading periods(.) and hypens(-).
Matches:	[my.domain.com], [regexlib.com], [big-reg.com]
Non-Matches:	[.mydomain.com], [regexlib.comm], [-bigreg.com]
Submitted By:	Stewart Haddock

Expression:	**^\d{4}[\-\/\s]?((((0[13578])|(1[02]))[\-\/\s]?(([0-2][0-9])|(3[01])))|(((0[469])|(11))[\-\/\s]?(([0-2][0-9])|(30)))|(02[\-\/\s]?[0-2][0-9]))$**
Description:	- validates a yyyy-mm-dd, yyyy mm dd, or yyyy/mm/dd date - makes sure day is within valid range for the month - does NOT validate Feb. 29 on a leap year, only that Feb. CAN have 29 days
Matches:	[0001-12-31], [9999 09 30], [2002/03/03]
Non-Matches:	[0001\02\30], [9999.15.01], [2002/3/3]
Submitted By:	Rex Po

Expression:	**^http\://[a-zA-Z0-9\-\.]+\.[a-zA-Z]{2,3}(/\S*)?$**
Description:	Verifies URLs. Checks for the leading protocol, a good looking domain (two or three letter TLD; no invalid characters in domain) and a somwhat reasonable file path.
Matches:	[http://psychopop.org], [http://www.edsroom.com/newUser.asp], [http://unpleasant.jarrin.net/markov/inde]
Non-Matches:	[ftp://psychopop.org], [http://www.edsroom/], [http://un/pleasant.jarrin.net/markov/index.asp]
Submitted By:	Klaxon Mindjammer

Expression:	**^([1-9]|[1-9]|0[1-9]|10|11|12)[0-5]\d$**
Description:	Matches a 12-hour time value expressed as either 4 numeric digits, 3 numeric digits, or a space and 3 numeric digits. 3 digit times (930) can be expressed with leading 0's (0930) or not. AM/PM designation is not included in this expression.
Matches:	[1145], [933], [801]
Non-Matches:	[0000], [1330], [8:30]
Submitted By:	Steven Smith

Expression:	**^\d{1,2}\/\d{2,4}$**
Description:	Accepts 1-2 digits followed by a slash followed by 2-4 digits. Useful for numeric month/year entry.
Matches:	[9/02], [09/2002], [09/02]
Non-Matches:	[Fall 2002], [Sept 2002]
Submitted By:	Mike Hensley

Expression:	**^(|(0[1-9])|(1[0-2]))\/((0[1-9])|(1\d)|(2\d)|(3[0-1]))\/((\d{4}))$**
Description:	This expression matches dates formatted as MM/DD/YYYY where months and days must be 2 digits each, zero padded. It is not perfect - it
Matches:	[01/01/2001], [02/30/2001], [12/31/2002]

Matches:	[01/01/2001], [02/30/2001], [12/31/2002]
Non-Matches:	[1/1/02], [1/1/2002], [1/25/2002]
Submitted By:	Steven Smith

Expression:	**^(1?(-?\d{3})-?)?(\d{3})(-?\d{4})$**
Description:	US Telephone Reg expression that allows 7, 10 or 11 digits with or without hyphens.
Matches:	[15615552323], [1-561-555-1212], [5613333]
Non-Matches:	[1-555-5555], [15553333], [0-561-555-1212]
Submitted By:	jay gann

Expression:	**<[^>]*name[\s]*=[\s]*"?[^\w_]*"?[^>]*>**
Description:	This RX is used to find get all named tags in an html string. If you find a problem with it, please email drew@lordstrange.com
Matches:	[<input type = text name = "bob">], [<select name = "fred">], [<form]
Non-Matches:	[<input type = submit>], [<font face = "arial">], [The drity brown fox stank like]
Submitted By:	Andrew Freese

Expression:	**^(1|01|2|02|3|03|4|04|5|05|6|06|7|07|8|08|9|09|10|11|12{1,2}):(([0-5]{1}[0-9]{1}\s{0,1})([AM|PM|am|pm]{2,2}))\W{0}$**
Description:	Validate time as h:mm AM/PM
Matches:	[1:00 AM], [12:00 PM], [1:00am]
Non-Matches:	[24:00]
Submitted By:	Jim Compton

Expression:	**^\d*$**
Description:	Accepts an unsigned integer number. Also matches empty strings.
Matches:	[123], [000], [43]
Non-Matches:	[asbc], [-34], [3.1415]
Submitted By:	Steven Smith

Expression:	**^[-+]?\d*$**
Description:	Matches any integer number or numeric string, including positive and negative value characters (+ or -). Also matches empty strings.
Matches:	[123], [-123], [+123]
Non-Matches:	[abc], [3.14159], [-3.14159]
Submitted By:	Steven Smith

Expression:	**^\d*\.?\d*$**
Description:	Matches any unsigned floating point number/numeric string. Also matches empty strings.
Matches:	[123], [3.14159], [.234]
Non-Matches:	[abc], [-3.14159], [3.4.2]
Submitted By:	Steven Smith

Expression:	**^((\d{5}-\d{4})|(\d{5})|([A-Z]\d[A-Z]\s\d[A-Z]\d))$**
Description:	This expression matches three different formats of postal codes: 5 digit US ZIP code, 5 digit US ZIP code + 4, and 6 digit alphanumeric Canadian Postal Code. The first one must be 5 numeric digits. The ZIP+4 must be 5 numeric digits, a hyphen, and then ...
Matches:	[44240], [44240-5555], [T2P 3C7]
Non-Matches:	[44240ddd], [t44240-55], [t2p3c7]
Submitted By:	Qing Jiang

Expression:	**^[\\(]{0,1}([0-9]){3}[\\)]{0,1}[ ]?([^0-1]){1}([0-9]){2}[ ]?[-]?[ ]?([0-9]){4}[ ]*((x){0,1}([0-9]){1,5}){0,1}$**
Description:	US Telephone Number where this is regular expression excludes the first number, after the area code,from being 0 or 1; it also allows an extension to be added where it does not have to be prefixed by 'x'.
Matches:	[(910)456-7890], [(910)456-8970 x12], [(910)456-8970 1211]
Non-Matches:	[(910) 156-7890], [(910) 056-7890], [(910) 556-7890 x]
Submitted By:	Bill Anderson

Expression:	**^((0?[1-9]|[12][1-9]|3[01])\.(0?[13578]|1[02])\.20[0-9]{2}|(0?[1-9]|[12][1-9]|30)\.(0?[13456789]|1[012])\.20[0-9]{2}|(0?[1-9]|1[1-9]|2[0-8])\.(0?[123456789]|1[012])\.20[0-9]{2}|(0?[1-9]|[12][1-9])\.(0?[123456789]|1[012])\.20(00|04|08|12|16|20|24|28|32|36|40|44|48|52|56|60|64|68|72|76|80|84|88|92|96))$**
Description:	Fully functional date validator in format dd.MM.yyyy Works only within range of years 2000-2099 ! It allows leading zeros but does not require them. The last year pattern (enumeration) is not very clever but I will improve it, if needed.
Matches:	[31.01.2002], [29.2.2004], [09.02.2005]
Non-Matches:	[31.11.2002], [29.2.2002], [33.06.2000]
Submitted By:	Dalibor Kalna

Expression:	**^(0[1-9]|1[0-2])\/((0[1-9]|2\d)|3[0-1])\/(19\d\d|200[0-3])$**
Description:	This expression is an attempt to match the most obvious features of a valid date in American/US format of the form mm/dd/yyyy for any year 1900 through 2003. It can easily be adapted for different date ranges. It matches any date where the day part is ...
Matches:	[12/31/2003], [01/01/1900], [11/31/2002]
Non-Matches:	[1/1/2002], [01/01/02], [01/01/2004]

Submitted By:	Josh Bailyr
Expression:	**^(((((13578])|(1[0-2]))[\-\/\s]?(([1-9])|([1-2][0-9])|(3[01])))|((([469])|(11))[\-\/\s]?(([1-9])|([1-2][0-9])|(30)))|(2[\-\/\s]?(([1-9])|([1-2][0-9])))))[\-\/\s]?\d{4})(\s((([1-9])|(1[02]))\:([0-5][0-9])((\s)|(\:([0-5][0-9])\s))([AM|PM|am|pm]{2,2})))?$**
Description:	This expression can be used validate a datetime column from SQL Server. Big parts of it where taken from other samples on RegexLib. Please feel free to take it apart and improve it.
Matches:	[3/3/2003], [3/3/2002 3:33 pm], [3/3/2003 3:33:33 am]
Non-Matches:	[13/1/2002], [3/3/2002 3:33], [31/3/2002]
Submitted By:	Scott Watermasysk Watermasysk

Expression:	**([a-zA-Z]:(\\w+)*\\[a-zA-Z0_9]+)?.xls**
Description:	This RegEx will help to validate a physical file path with a specific file extension (here xls)
Matches:	[E:\DyAGT\SD01A_specV2.xls]
Non-Matches:	[E:\DyAGT\SD01A_specV2.txt]
Submitted By:	vinod kumar

Expression:	**(((0[13578]|10|12)([-./])(0[1-9]|[12][0-9]|3[01])([-./])(\d{4}))|((0[469]|11)([-./])([0][1-9]|[12][0-9]|30)([-./])(\d{4}))|((2)([-./])(0[1-9]|1[0-9]|2[0-8])([-./])(\d{4}))|((2)(\.|-|\/)(29)([-./])([02468][048]00))|((2)([-./])(29)([-./])([13579][26]00))|((2)([-./])(29)([-./])([0-9][0-9][0][48]))|((2)([-./])(29)([-./])([0-9][0-9][2468][048]))|((2)([-./])(29)([-./])([0-9][0-9][13579][26]))**
Description:	My meager attempt at a date validator with leap years using a strict mm/dd/yyyy format.
Matches:	[02/29/2084], [01/31/2000], [11/30/2000]
Non-Matches:	[02/29/2083], [11/31/2000], [01/32/2000]
Submitted By:	Jason West

Expression:	**^[a-zA-Z0-9\s.\-]+$**
Description:	ANY alphanumeric string with spaces, commas, dashes.
Matches:	[2222 Mock St.], [1 A St.], [555-1212]
Non-Matches:	[[A Street]], [(3 A St.)], [{34 C Ave.}]
Submitted By:	Mart Maasikas

Expression:	**Last.*?(\d+.?\d*)**

Non-Matches:	[[AADDSS]]
Submitted By:	Prasad DV

Expression:	**^([0-9](|-)?)?(\(?[0-9]{3}\)?|[0-9]{3})(|-)?([0-9]{3}(|-)?[0-9]{4}|[a-zA-Z0-9]{7})$**
Description:	Matches US phone number format. 1 in the beginning is optional, area code is required, spaces or dashes can be used as optional divider between number groups. Also alphanumeric format is allowed after area code.
Matches:	[1-(123)-123-1234], [123 123 1234], [1-800-ALPHNUM]
Non-Matches:	[1.123.123.1234], [(123)-1234-123], [123-1234]
Submitted By:	Igor Kravtsov

Expression:	**^([0-1][0-9]|[2][0-3]):([0-5][0-9])$**
Description:	Matches a string if it is a valid time in the format of HH:MM
Matches:	[02:04], [16:56], [23:59]
Non-Matches:	[02:00 PM], [PM2:00], [24:00]
Submitted By:	Koen Hoorelbeke

Expression:	**^[0,1]?\d{1}\/(([0-2]?\d{1})|([3][0,1]{1}))\/(([1]{1}[9]{1}[9]{1}\d{1})|([2-9]{1}\d{3}))$**
Description:	This expression checks the validity of a date (US, but it is easily editable for other format's). Year's 1990-9999, Month's 1 or 01 to 12, Day's 1 or 01 to 31. Still needs to have individual months added (i.e., Feb's 28 days), and some how to check for l ...
Matches:	[01/01/1990], [12/12/9999], [3/28/2001]
Non-Matches:	[3-8-01], [13/32/1001], [03/32/1989]
Submitted By:	Scott Watermasysk

Expression:	**((\(\d{3}\)?)|(\d{3}))([\s-./]?)(\d{3})([\s-./]?)(\d{4})**
Description:	US Phone number that accept a dot, a space, a dash, a forward slash, between the numbers. Will Accept a 1 or 0 in front. Area Code not necessary
Matches:	[1.2123644567], [0-234.567/8912], [1-(212)-123 4567]
Non-Matches:	[0-212364345], [1212-364,4321], [0212\345/6789]
Submitted By:	Mart Maasikas

Expression:	**^([0-9]{6}[\s\-]{1}[0-9]{12}|[0-9]{18})$**
Description:	This regular expression matches 'Switch' card numbers - a payment method used extensively in the UK.
Matches:	[000000 0000000000000], [000000-0000000000000], [0000000000000000000]
Non-Matches:	[000000_0000000000000]
Submitted By:	Scott Frampton

Expression:	**^(([1-9])|(0[1-9])|(1[0-2]))\/((0[1-9])|([1-31]))\/((\d{2})|(\d{4}))$**
Description:	Matches U.S. dates with leading zeros and without and with 2 or four digit years

Description:	Matches U.S. dates with leading zeros and without and with 2 or four digit years
Matches:	[01/01/2001], [1/1/2001], [01/1/01]
Non-Matches:	[13/01/2001], [1/2/100], [09/32/2001]
Submitted By:	Chris Becker

Expression:	**^\$?([0-9]{1,3},([0-9]{3},)*[0-9]{3}|[0-9]+)(.[0-9][0-9])?$**
Description:	Matches currency input with or without commas.
Matches:	[$3,023,123.34], [9,876,453], [123456.78]
Non-Matches:	[4,33,234.34], [$1.234], [abc]
Submitted By:	Brian Orrell

Expression:	**^\d{5}$\|^\d{5}-\d{4}$**
Description:	This regular expression will match either a 5 digit ZIP code or a ZIP+4 code formatted as 5 digits, a hyphen, and another 4 digits. Other than that, this is just a really really long description of a regular expression that I'm using to test how my fron ...
Matches:	[55555-5555], [34564-3342], [90210]
Non-Matches:	[434454444], [645-32-2345], [abc]
Submitted By:	Steven Smith

Expression:	**^\d{3}-\d{2}-\d{4}$**
Description:	This regular expression will match a hyphen-separated Social Security Number (SSN) in the format NNN-NN-NNNN.
Matches:	[333-22-4444], [123-45-6789]
Non-Matches:	[123456789], [SSN]
Submitted By:	Steven Smith

Expression:	**^[2-9]\d{2}-\d{3}-\d{4}$**
Description:	This expression matches a hyphen separated US phone number, of the form ANN-NNN-NNNN, where A is between 2 and 9 and N is between 0 and 9.
Matches:	[800-555-5555], [333-444-5555], [212-666-1234]
Non-Matches:	[000-000-0000], [123-456-7890], [2126661234]
Submitted By:	Steven Smith

Expression:	**^\d{5}-\d{4}|\d{5}|[A-Z]\d[A-Z] \d[A-Z]\d$**
Description:	This expression matches three different formats of postal codes: 5 digit US ZIP code, 5 digit US ZIP code + 4, and 6 digit alphanumeric Canadian Postal Code. The first one must be 5 numeric digits. The ZIP+4 must be 5 numeric digits, a hyphen, and then ...
Matches:	[44240], [44240-5555], [G3H 6A3]
Non-Matches:	[Ohio], [abc], [g3h6a3]
Submitted By:	Steven Smith

Expression:	**[0-9]{4}\s*[a-zA-Z]{2}**
Description:	Dutch zip code expression 4 numbers - space yes/no - 2 letters
Matches:	[1054 WD], [1054WD], [1054 wd]
Non-Matches:	[10543]
Submitted By:	Roland Mensenkamp

Expression:	**(^1300\d{6}$)\|(^1800\|1900\|1902\d{6}$)|(^0[2|3|7|8]{1}[0-9]{8}$)\|(^13\d{4}$)|(^04\d{2,3}\d{6}$)**
Description:	Australian phone numbers: Matches all known formats incl normal 10-digit landline numbers (valid area code mandatory) 13, 1300, 1800, 1900, 1902 plus mobile 10 and 11-digit formats. Use a Replace function first to remove non-numerics which are probably s ...
Matches:	[0732105432], [1300333444], [131313]
Non-Matches:	[32105432], [13000456]
Submitted By:	Bill Hely

Expression:	**^((https?|ftp)\://(((\[?(\d{1,3}\.){3}\d{1,3}\]?)|(([-a-zA-Z0-9]+\.)+[a-zA-Z]{2,4}))(\:\d+)?(/[-a-zA-Z0-9._?,'+&%$#=~\\]+)*/?)$**
Description:	Using other regular experssions from this page, combining others for email addresses, and mixing in my own ideas - I came up with this regular expression. Can be used to validate input into a database.
Matches:	[http://207.68.172.254/home.ashx], [ftp://ftp.netscape.com/], [https://www.brinkster.com/login.asp]
Non-Matches:	[htp://mistake.com/], [http://www_address.com/], [ftp://www.files.com/file with spaces.txt]
Submitted By:	Benjamin Gray

Expression:	**([0-9]{4})-([0-9]{1,2})-([0-9]{1,2})**
Description:	Checks for Date in the typical MySQL DB Format. Not mutch but simple to for converting to German date format: $date = "2003-12-03"; if (ereg ("([0-9]{4})-([0-9]{1,2})-([0-9]{1,2})", $date, $regs)) { echo "$regs[3].$reg ...
Matches:	[2002-11-03], [2007-17-08], [9999-99-99]
Non-Matches:	[2002/17/18], [2002.18.45], [18.45.2002]
Submitted By:	Mike T Spike

Expression:	**^\$?(\d{1,3}(\,\d{3})*|(\d+))(\.\d{0,2})?$**
Description:	From Author: DON'T USE THIS ONE. FIND MY OTHER ONE THAT BLOCKS LEADING ZEROS. My site also couldn't swallow the \d, so I switched to numeric ranges and it worked fine. KEYWORDS Currency Money Dollar
Matches:	[$0,234.50], [0234.5], [0,234.]
Non-Matches:	[$1,23,50], [$123.123]
Submitted By:	Tom Persing

Expression:	**(^\d{5}-\d{3}|^\d{2}.\d{3}-\d{3}|\d{8})**
Description:	Validar o do CEP Brasileiro com 8 posicoes podendo usar mascara . e - ou somente numeros

Matches:	[12.345-678], [12345-678], [12345678]
Non-Matches:	[12.345678], [12345-1], [123]
Submitted By:	Fernando Cerqueira

Expression:	**^([a-zA-Z]\:|\\)\\([^\\]+\\)*[^\/:*?"<>|]+\.htm(l)?$**
Description:	Validates a file path on your local drive or a network drive. A similar one was written by Vinod Kumar but it does not reject asterisks in the path. Moreover, his did not work with RegularExpressionValidators. This one does.
Matches:	[x:\test\testing.htm], [x:\test\test#$ ing.html], [\\test\testing.html]
Non-Matches:	[x:\test\test/ing.htm], [x:\test\test*.htm], [\\test?<.htm]
Submitted By:	asdf asdfa

Expression:	**^[1-9]{1}[0-9]{3}$**
Description:	Postcode for Belgium
Matches:	[1234]
Non-Matches:	[123], [123A]
Submitted By:	Henk de Vries -op regexlib.com-

Expression:	**^[A-Z]{1}(|-)?[1-9]{1}[0-9]{3}$**
Description:	Postcode for Germany
Matches:	[A-1234], [A 1234], [A1234]
Non-Matches:	[AA-1234], [A12345]
Submitted By:	Henk de Vries -op regexlib.com-

Expression:	**^(F-)?[0-9]{5}$**
Description:	Postcode check for France (including colonies)
Matches:	[12345], [F-12345]
Non-Matches:	[F12345], [F-123456], [123456]
Submitted By:	Henk de Vries -op regexlib.com-

Expression:	**^(V-|I-)?[0-9]{4}$**
Description:	Postcode check for Italy (including possible Vatican/Italy indications)
Matches:	[1234], [V-1234]
Non-Matches:	[12345]
Submitted By:	Henk de Vries -op regexlib.com-

Expression:	**^[1-9]{1}[0-9]{3} ?[A-Z]{2}$**
Description:	Postcode check for Netherlands
Matches:	[1234 AB], [1234AB]

Matches:	[1234 AB], [1234AB]
Non-Matches:	[123AB], [1234AAA]
Submitted By:	Henk de Vries -op regexlib.com-

Expression:	**^([1-9]{2}|[0-9][1-9]|[1-9][0-9])[0-9]{3}$**
Description:	Postcode check for Spain
Matches:	[12345], [10234], [01234]
Non-Matches:	[00123]
Submitted By:	Henk de Vries -op regexlib.com-

Expression:	**^(/w|/W|[^<>+?$%{}&])+$**
Description:	simple expression for excluding a given list of characters. simply change the contents of [^] to suite your needs. for example ^(/w|/W|[^<>])+$ would allow everything except the characters < and >.
Matches:	[John Doe Sr.], [100 Elm St., Suite 25], [Valerie's Gift Shop]
Non-Matches:	[Hey]
Submitted By:	Glenn Caccia

Expression:	**<[a-zA-Z][^>]*\son\w+=(\w+|'[^']*'|"[^"]*")[^>]*>**
Description:	Find HTML tags that have javascript events attached to them.
Matches:	[<IMG onmouseover="window.close()">]
Non-Matches:	[<IMG src="star.gif">]
Submitted By:	Lewis Moten

Expression:	**(?!^0*$)(?!^0*\.0*$)^\d{1,5}(\.\d{1,3})?$**
Description:	This regular expression validates a number NOT 0, with no more than 5 places ahead and 3 places behind the decimal point.
Matches:	[1], [12345.123], [0.5]
Non-Matches:	[0], [0.0], [123456.1234]
Submitted By:	Michael Trefry

Expression:	**^.+@[^\.].*\.[a-z]{2,}$**
Description:	Most email validation regexps are outdated and ignore the fact that domain names can contain any foreign character these days, as well as the fact that anything before @ is acceptable. The only roman alphabet restriction is in the TLD, which for a long t ...
Matches:	[whatever@somewhere.museum], [foreignchars@myforeigncharsdomain.nu], [me+mysomething@mydomain.com]
Non-Matches:	[a@b.c], [me@.my.com], [a@b.comFOREIGNCHAR]
Submitted By:	Thor Larholm

Expression:	**^(\d{5}-\d{4}|\d{5})$**
Description:	this works with ASP.net regular expression valiadtors, ecma script compliant

Matches:	[12345], [12345-1234]
Non-Matches:	[12345-12345], [123], [12345-abcd]
Submitted By:	lynn eriksen

Expression:	**^(\d{1,2}|1\d\d|2[0-4]\d|25[0-5])\.(\d{1,2}|1\d\d|2[0-4]\d|25[0-5])\.(\d{1,2}|1\d\d|2[0-4]\d|25[0-5])\.(\d{1,2}|1\d\d|2[0-4]\d|25[0-5])$**
Description:	This matches an IP address, putting each number in its own group that can be retrieved by number. If you do not care about capturing the numbers, then you can make this shorter by putting everything after ^ until immediately after the first \. in a group ...
Matches:	[0.0.0.0], [255.255.255.02], [192.168.0.136]
Non-Matches:	[256.1.3.4], [023.44.33.22], [10.57.98.23.]
Submitted By:	Andrew Polshaw

Expression:	**<img([^>]*[^/])>**
Description:	Use this along with this replacement string <img\1 /> to convert image tags to XHTML compliant image tags.
Matches:	[<img src="bob">]
Non-Matches:	[]
Submitted By:	Murray Roke

Expression:	**<!--[\s\S]*?-->**
Description:	Removes pesky comments and commented javascript from HTML
Matches:	[<!-- comments -->], [<!-- x = a > b - 3 -->]
Non-Matches:	[<COMMENTS>this is a comment</COMMENTS>]
Submitted By:	Lewis Moten

Expression:	**</?(\w+)(\s+\w+=(\w+|"[^"]*"|'[^']*'))*>**
Description:	Finds any HTML tag and sub-matches properties weather it has an apposterphee, quote, or no quote/apposterphee
Matches:	[<TD>], [<TD bgColor="FFFFFF">], [</TD>]
Non-Matches:	[No Tag Here ...]
Submitted By:	Lewis Moten

Expression:	**(\{\\f\d*)\\([^;]+;)**
Description:	This pattern returns the font section from an RTF document. The first parenthetical subexpression captures the font number, the second returns the actual font enumeration. Lame-o, but fun! :-)
Matches:	[{\f0\Some Font names here;], [{\f1\fswiss\fcharset0\fprq2{*\panose 020b0604020202020204}Arial;], [{\f]
Non-Matches:	[{f0fs20 some text}]
Submitted By:	Joel Dooris

Expression:	**</?([a-zA-Z][-A-Za-z\d\.]{0,71})(\s+(\S+)(\s*=\s*([-**

	\w\.]{1,1024}|"[^"]{0,1024}"|'[^']{0,1024}'))?)*\s*>
Description:	Searches for tags and there atributes according to the HTML 2.0 specification to limit length of tags to 72 characters, and length of attribute values to 1024 characters.
Matches:	[<IMG src='stars.gif' alt="space" height=1>]
Non-Matches:	[this is not a tag]
Submitted By:	Lewis Moten
Expression:	**^([0]\d|[1][0-2])\/([0-2]\d|[3][0-1])\/([2][01]|[1][6-9])\d{2}(\s([0-1]\d|[2][0-3])(\:[0-5]\d){1,2})?$**
Description:	This is a regular expression to validate a date string in "MM/DD/YYYY" format, a date time string in "MM/DD/YYYY HH:MM" or a date time string in "MM/DD/YYYY HH:MM:SS" format. It can validate date from 1600 to 2199.
Matches:	[12/30/2002], [01/12/1998 13:30], [01/28/2002 22:35:00]
Non-Matches:	[13/30/2002], [01/12/1998 24:30], [01/28/2002 22:35:64]
Submitted By:	Spring Zhang
Expression:	**((?<strElement>(^[A-Z0-9-;=]*:))(?<strValue>(.*)))**
Description:	Usesful for importing vcards. Matches vcard elements and values.
Matches:	[BEGIN:], [TEL;WORK;VOICE:], [TEL:]
Non-Matches:	[begin:], [TEL;PREF;]
Submitted By:	Karsten Grombach
Expression:	**^<a\s+href\s*=\s*"http:\/\/([^"]*)"([^>]*)>(.*?(?=<\/a>))<\/a>$**
Description:	Regexp to find all external links in a HTML string. Can easily be modified to handle all/other links/protocols (like file/https/ftp). Uses lookahead assertions and non-greedy modifier to check for the end but still allow html tags inbetw ...
Matches:	[my external link], [<a href="http:/]
Non-Matches:	[my internal link]
Submitted By:	Anders Rask
Expression:	**^([0]\d|[1][0-2])\/([0-2]\d|[3][0-1])\/([2][01]|[1][6-9])\d{2}(\s([0]\d|[1][0-2])(\:[0-5]\d){1,2})*\s*([aApP][mM]{0,2})?$**
Description:	This is a combination of a few regular expressions found on this site. It allows for a flexible date and time combination, but requires a 12-hour clock (am/pm). Many versions of the am/pm are supported.
Matches:	[12/31/2002], [12/31/2002 08:00], [12/31/2002 08:00 AM]
Non-Matches:	[12/31/02], [12/31/2002 14:00]
Submitted By:	Bill LaPrade
Expression:	**<blockquote>(?:\s*([^<]+) \s*)+</blockquote>**

	<blockquote>string1 string2 string3 </blockquote>
Matches:	[<blockquote>string1 string2 string3 </blockquote>]
Non-Matches:	[..]
Submitted By:	Waheed Khan

Expression:	**^((0?[13578]|10|12)(-|\/)((0[0-9])|([12])([0-9]?)|(3[01]?))(-|\/)((19)([2-9])(\d{1})|(20)([01])(\d{1})|([8901])(\d{1}))|(0?[2469]|11)(-|\/)((0[0-9])|([12])([0-9]?)|(3[0]?))(-|\/)((19)([2-9])(\d{1})|(20)([01])(\d{1})|([8901])(\d{1}))$**
Description:	date validation MM/DD/YYYY or MM/DD/YY: 1/1/1920 through 12/31/2019; Feb 30 is always allowed
Matches:	[1/2/03], [2/30/1999], [03/04/19]
Non-Matches:	[3/4/2020], [3/4/1919]
Submitted By:	Brad Williams

Expression:	**</?(\w+)(\s*\w*\s*=\s*("[^"]*"|'[^']'|[^>]*))*|/?>**
Description:	Matches HTML of XML tags, with or without attributes (single-, double-, or non-quoted), closing tags, or self-closing singleton tags.
Matches:	[], [], [ ]
Non-Matches:	[this is a test...]
Submitted By:	Jean-Philip Losier

Expression:	**^ *(1[0-2]|[1-9]):[0-5][0-9] *(a|p|A|P)(m|M) *$**
Description:	validate 12-hour time with am/pm after it, with optional spaces before or after, and optionally between time and am/pm.
Matches:	[12:00am], [1:00 PM], [12:59 pm]
Non-Matches:	[0:00], [0:01 am], [13:00 pm]
Submitted By:	Jim Tebbel

Expression:	**\({1}[0-9]{3}\){1}\-{1}[0-9]{3}\-{1}[0-9]{4}**
Description:	
Matches:	[(111)-111-1111]
Non-Matches:	[11111111111]
Submitted By:	s4 s

Expression:	**[^abc]**
Description:	
Matches:	[def]
Non-Matches:	[abc]
Submitted By:	Krisada Arjinpattara

Expression:	**^(([0]?[1-9]|[1][0-2])[\/|\-|\.]([0-2]\d|[3][0-1]|[1-9])[\/|\-|\.]([2][0])?\d{2}\s+((([0][0-9]|[1][0-2]|[0-9])[\:|\-|\.]([0-5]\d)\s*([aApP][mM])?)|(([0-1][0-9]|[2][0-3]|[0-9])[\:|\-|\.]([0-5]\d))))$**
Description:	MM-DD-20YY HH:MM AP MM-DD-20YY H:MM AP MM-DD-20YY HH:MM MM-DD-20YY H:MM MM-DD-YY HH:MM AP MM-DD-YY H:MM AP MM-DD-YY HH:MM MM-DD-YY H:MM M-D-YY HH:MM AP M-D-YY H:MM AP M-D-YY HH:MM M-D-YY H:MM where - can be / or - or . where : can be : or ...
Matches:	[01/01/2002 04:42], [5-12-02 04:42 AM], [01.01/02 04-42aM]
Non-Matches:	[01-12-1999 4:50PM], [01-12-2002 15:10PM], [01-12-002 8:20PM]
Submitted By:	Bob Duffett

Expression:	**^([1][12]|[0]?[1-9])[\/-]([3][01]|[12]\d|[0]?[1-9])[\/-](\d{4}|\d{2})$**
Description:	Matches dates with the following format Month - either MM (like 01) or M (like 1) - from 1 to 12 Day - either DD (like 01) or D (like 1) - from 1 to 31 Year - either YYYY (like 1998) or YY (like 98) Separater - either - or /
Matches:	[11-02-02], [1-25-2002], [01/25/2002]
Non-Matches:	[13-02-02], [11.02.02], [11/32/2002]
Submitted By:	Mekonnen Mekonnen

Expression:	**(([0-1][0-9])|([2][0-3])):([0-5][0-9]):([0-5][0-9])**
Description:	Validates time in MySQL time format. 24 hour time colon seperated hours:minutes:seconds
Matches:	[09:30:00], [17:45:20], [23:59:59]
Non-Matches:	[24:00:00]
Submitted By:	Patrick O'Hara

Expression:	**(((0[1-9]|[12][0-9]|3[01])([-./])(0[13578]|10|12)([-./])(\d{4}))|(([0][1-9]|[12][0-9]|30)([-./])(0[469]|11)([-./])(\d{4}))|((0[1-9]|1[0-9]|2[0-8])([-./])(02)([-./])(\d{4}))|((29)(\.|-|\/)(02)([-./])([02468][048]00))|((29)([-./])(02)([-./])([13579][26]00))|((29)([-./])(02)([-./])([0-9][0-9][0][48]))|((29)([-./])(02)([-./])([0-9][0-9][2468][048]))|((29)([-./])(02)([-./])([0-9][0-9][13579][26]))))**
Description:	Jason West (jason.west@mail.state.ky.us) date validator with leap years using a strict dd/mm/yyyy (ITALIAN) format
Matches:	[29/02/2000], [31/01/2000], [30-01-2000]
Non-Matches:	[29/02/2002], [32/01/2002], [10/2/2002]
Submitted By:	Alessandro Olivi

Expression:	**^0[1-6]{1}(([0-9]{2}){4})|((\s[0-9]{2}){4})|((-[0-9]{2}){4})$**
Description:	Regular Expression that validate phone in France.
Matches:	[01 46 70 89 12], [01-46-70-89-12], [0146708912]
Non-Matches:	[01-46708912], [01 46708912], [+33235256677]

Submitted By:	Bertrand Perrin
Expression:	**^[0-9A-Za-z_]+(.[jJ][pP][gG]|.[gG][iI][fF])$**
Description:	Matches HTML image leaf filenames.
Matches:	[good.gif], [go d.GIf], [goo_d.jPg]
Non-Matches:	[junk], [bad.bad.gif], [slash\gif.]
Submitted By:	Geoffrey Swenson
Expression:	**<[^>\s]*\bauthor\b[^>]*>**
Description:	This expression will match the corresponding XML/HTML elements opening and closing tags. Useful to handle documents fragments, without loading an XML DOM.
Matches:	[<author name="Daniel">], [</sch:author>], [<pp:author name="Daniel"]
Non-Matches:	[<other>], [</authors>], [<work>author</work>]
Submitted By:	Daniel Cazzulino
Expression:	**^(?:(?:(?:(?:1[6-9]|[2-9]\d)?(?:0[48]|[2468][048]|[13579][26])|(?:(?:16|[2468][048]|[3579][26])00)))(\/|-|\.)(?:0?2\1(?:29))$)|(?:(?:1[6-9]|[2-9]\d)?\d{2})(\/|-|\.)(?:(?:(?:0?[13578]|1[02])\2(?:31))|(?:(?:0?[1,3-9]|1[0-2])\2(29|30))|(?:(?:0?[1-9])|(?:1[0-2]))\2(?:0?[1-9]|1\d|2[0-8]))$**
Description:	This expression validates dates in the y/m/d format from 1600/1/1 - 9999/12/31. Follows the same validation rules for dates as my other date validator (m/d/y format) located in this library.
Matches:	[04/2/29], [2002-4-30], [02.10.31]
Non-Matches:	[2003/2/29], [02.4.31], [00/00/00]
Submitted By:	Michael Ash
Expression:	**(\d*)'*-*(\d*)/*(\d*)"**
Description:	This regular expression is for parsing feet and inches measurements.
Matches:	[5'-3/16"], [1'-2"], [5/16"]
Non-Matches:	[1 3/16]
Submitted By:	Felix Osegueda
Expression:	**^[1-9]{1}$|^[1-4]{1}[0-9]{1}$|^50$**
Description:	Match any number between 1 and 50, no characters, no empty sets, and not zero. Match numbers greater than 1 and less than 51, no spaces, no characters.
Matches:	[1], [23], [50]
Non-Matches:	[0], [111], [xyz]
Submitted By:	Michael Gaertner

Expression:	**^([\u00c0-\u01ffa-zA-Z'])+$**
Description:	Expression to match names and dis-allow any attempts to send evil characters. In particular, it tries to allow non-english names by allowing unicode characters.
Matches:	[Jon Doe], [Jørn], [Mc'Neelan]
Non-Matches:	[Henry); hacking attempt]
Submitted By:	tom ferguson

Expression:	**^((([0]?[1-9]|1[0-2])(:|\.)(00|15|30|45)?( )?(AM|am|aM|Am|PM|pm|pM|Pm))|(([0]?[0-9]|1[0-9]|2[0-3])(:|\.)(00|15|30|45)?))$**
Description:	This expression will allow a time value with 15 minute increments. It will not accept seconds. You can use either US time or Military time format.
Matches:	[1:00 PM], [6:45 am], [17:30]
Non-Matches:	[4:32 am], [5:30:00 am], [17:01]
Submitted By:	Rick Yokum

Expression:	**(^\d*\.?\d*[1-9]+\d*$)\|(^[1-9]+\d*\.\d*$)**
Description:	Accepts only positive decimal values. Zero and negatvie numbers are non-matching. Allows zeros after last non-zero numeric value after decimal place for significant digits.
Matches:	[0.050], [5.0000], [5000]
Non-Matches:	[0], [0.0], [.0]
Submitted By:	Bri Gipson

Expression:	**^\{?[a-fA-F\d]{8}-([a-fA-F\d]{4}-){3}[a-fA-F\d]{12}\}?$**
Description:	Validates a GUID with and without brackets. 8,4,4,4,12 hex characters seperated by dashes.
Matches:	[{e02ff0e4-00ad-090A-c030-0d00a0008ba0}], [e02ff0e4-00ad-090A-c030-0d00a0008ba0]
Non-Matches:	[0xe02ff0e400ad090Ac0300d00a0008ba0]
Submitted By:	Lewis Moten

Expression:	**^([a-zA-Z0-9@*#]{8,15})$**
Description:	Password matching expression. Match all alphanumeric character and predefined wild characters. Password must consists of at least 8 characters and not more than 15 characters.
Matches:	[@12X*567], [1#Zv96g@*Yfasd4], [#67jhgt@erd]
Non-Matches:	[$12X*567], [1#Zv_96], [+678jhgt@erd]
Submitted By:	lawson law

Expression:	**^([0-1]([\s-./\\])?)?(\(?[2-9]\d{2}\)?|[2-9]\d{3})([\s-./\\])?([0-9]{3}([\s-./\\])?[0-9]{4}|[a-zA-Z0-9]{7}|([0-9]{3}[-][a-zA-Z0-9]{4}))**
Description:	USA Alhpanumeric Valid Phone numbers

Matches:	[1.222.333.1234], [1-223-123-1232], [1-888-425-DELL]
Non-Matches:	[1.1.123123.123], [12-1322-112-31], [1-800-CALL-DEL]
Submitted By:	Amit Deshpande

Expression:	**^([A-Z]{1}[a-z]{1,})$\|^([A-Z]{1}[a-z]{1,}\040[A-Z]{1}[a-z]{1,})$|^([A-Z]{1}[a-z]{1,}\040[A-Z]{1}[a-z]{1,}\040[A-Z]{1}[a-z]{1,})$\|^$**
Description:	Matches up to three alphabet words separated by spaces with first alphabet character of each word uppercase. Also matches empty strings.
Matches:	[Sacramento], [San Francisco], [San Luis Obispo]
Non-Matches:	[SanFrancisco], [SanLuisObispo], [San francisco]
Submitted By:	Don Batchelor

Expression:	**(("|')[a-z0-9\/\.\?\=\&]*(\.htm|\.asp|\.php|\.jsp)[a-z0-9\/\.\?\=\&]*("|'))|(href=*?[a-z0-9\/\.\?\=\&"']*)**
Description:	Will locate an URL in a web page. It'll search in 2 ways - first it will try to locate a href=, and then go to the end of the link. If there is nu href=, it will search for the end of the file instead (.asp, .htm and so on), and then take the data between ...
Matches:	[href="produktsida.asp?kategori2=218"], [href="NuclearTesting.htm"]
Non-Matches:	[U Suck]
Submitted By:	Henric Rosvall

Expression:	**^(((((0[1-9])|(1\d)|(2[0-8]))-((0[1-9])|(1[0-2])))|((31-((0[13578])|(1[02])))|((29|30)-((0[1,3-9])|(1[0-2])))))-((20[0-9][0-9]))|(29-02-20(([02468][048])|([13579][26])))$**
Description:	This expression validates a date field in the European DD-MM-YYYY format. Days are validate for the given month and year.
Matches:	[05-01-2002], [29-02-2004], [31-12-2002]
Non-Matches:	[1-1-02], [29-02-2002], [31-11-2002]
Submitted By:	Jörg Maag

Expression:	**^\d*[0-9](|.\d*[0-9]|,\d*[0-9])?$**
Description:	This is permit all decimal number, exclude all alphanumeric caracter
Matches:	[123456.123456], [123456,123456], [123456]
Non-Matches:	[123a.123], [123a,123], [a]
Submitted By:	Hugues Gauthier

Expression:	**^(ac|AC|al|AL|am|AM|ap|AP|ba|BA|ce|CE|df|DF|es|ES|go|GO|ma|MA|mg|MG|ms|MS|mt|MT|pa**

Submitted By:	Fernando Cerqueira
Expression:	**^[+]?\d*$**
Description:	This re was used for set numbers only! Somente numeros são requeridos!
Matches:	[0123456789], [1234], [1]
Non-Matches:	[1.0?&], [a1], [2a-]
Submitted By:	Ramon Durães
Expression:	**<[aA][ ]{0,}([a-zA-Z0-9"'_.:;!?@$&()%=/]\|[-]\|[\f]){0,}>((<(([a-zA-Z0-9"'_.:;!?@$&()%=/]|[-]|[\f]){0,})>([a-zA-Z0-9"'_.:;!?@$&()%=/]\|[-]\|[\f]){0,})\|(([a-zA-Z0-9"'_.:;!?@$&()%=/]|[-]|[\f]){0,})){0,}**
Description:	I wrote this sweet little (well, not so little really) reg to extract links from an HTML source.... it is very robust, give it a try. The only limitation I have discovered is that it can't match invalid HTML...
Matches:	[this text is italicized]
Non-Matches:	[<A href='#'><P</A></P>]
Submitted By:	Brian Webb
Expression:	**[A-Z][a-z]+**
Description:	This expression was developed to match the Title cased words within a Camel cased variable name. So it will match 'First' and 'Name' within 'strFirstName'.
Matches:	[strFirstName], [intAgeInYears], [Where the Wild Things Are]
Non-Matches:	[123], [abc], [this has no caps in it]
Submitted By:	Steven Smith
Expression:	**(^-\d*\.?\d*[1-9]+\d*$)\|(^-[1-9]+\d*\.\d*$)**
Description:	Accepts only negative decimal values. Zero and positive numbers are non-matching. Allows zeros after last non-zero numeric value after decimal place for significant digits.
Matches:	[-0.050], [-5.000], [-5]
Non-Matches:	[0], [0.0], [.0]
Submitted By:	Bri Gipson
Expression:	**^([2][0]\d{2}\/([0]\d|[1][0-2])\/([0-2]\d|[3][0-1]))$|^([2][0]\d{2}\/([0]\d|[1][0-2])\/([0-**
Description:	this is a regular expression that for chinese programer! yyyy/mm/dd hh:mm:ss
Matches:	[2002/02/03], [2002/02/03 12:12:18]
Non-Matches:	[2002/02/36], [02/03/2002]
Submitted By:	martix wang

Submitted By:	martix wang

Expression:	**^(\d|,)*\.?\d*$**
Description:	Matches Numeric with Commas and a single decimal point. Also matches empty string.
Matches:	[1,000], [3,000.05], [5,000,000]
Non-Matches:	[abc], [$100,000], [Forty]
Submitted By:	Kevin Read

Expression:	**^\d$**
Description:	Matches exactly 1 numeric digit (0-9).
Matches:	[1], [2], [3]
Non-Matches:	[a], [324], [num]
Submitted By:	Steven Smith

Expression:	**^[0-9]+$**
Description:	Validate a string to see if it contains a number / integer
Matches:	[1234567890], [1234567890], [1234567890]
Non-Matches:	[http://none], [http://none], [http://none]
Submitted By:	Mr M

Expression:	**^.{4,8}$**
Description:	Matches any string between 4 and 8 characters in length. Limits the length of a string. Useful to add to password regular expressions.
Matches:	[asdf], [1234], [asdf1234]
Non-Matches:	[asd], [123], [asdfe12345]
Submitted By:	Steven Smith

Expression:	**^[\w\.=-]+@[\w\.-]+\.[\w]{2,3}$**
Description:	Much simpler email expression. This one forces a length of 2 or 3, which fits current specs, but you may need to alter the end as this one allows all numerals on the .COM section.
Matches:	[a@a.com], [a@a.com.au], [a@a.au]
Non-Matches:	[word], [word@], [@word]
Submitted By:	Gregory Beamer

Expression:	**^(\+|-)?\d+$**
Description:	Matches any signed integer.
Matches:	[-34], [34], [+5]

Matches:	[-34], [34], [+5]
Non-Matches:	[abc], [3.1415], [-5.3]
Submitted By:	Steven Smith

Expression:	**^[12345]$**
Description:	This matches a single numeric digit between 1 and 5, and is the same as saying ^[1-5]$.
Matches:	[1], [2], [4]
Non-Matches:	[6], [-1], [abc]
Submitted By:	Steven Smith

Expression:	**^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$**
Description:	This expression matches email addresses, and checks that they are of the proper form. It checks to ensure the top level domain is between 2 and 4 characters long, but does not check the specific domain against a list (especially since there are so many ...
Matches:	[joe@aol.com], [joe@wrox.co.uk], [joe@domain.info]
Non-Matches:	[a@b], [notanemail], [joe@@.]
Submitted By:	Steven Smith

Expression:	**^\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,3}$**
Description:	Simple email expression. Doesn't allow numbers in the domain name and doesn't allow for top level domains that are less than 2 or more than 3 letters (which is fine until they allow more). Doesn't handle multiple "." in the domain (joe@abc.co ...
Matches:	[joe@aol.com], [ssmith@aspalliance.com], [a@b.cc]
Non-Matches:	[joe@123aspx.com], [joe@web.info], [joe@company.co.uk]
Submitted By:	Steven Smith

Expression:	**^\d{5}$**
Description:	Matches 5 numeric digits, such as a zip code.
Matches:	[33333], [55555], [23445]
Non-Matches:	[abcd], [1324], [as;lkjdf]
Submitted By:	Steven Smith

Expression:	**(\w+)\s+\1**
Description:	This expression uses a BackReference to find occurrences of the same word twice in a row (separated by a space). Matches things like 'mandate dated', which may not be desirable. See Sean Carley's update for a better expression for true repeated word ...
Matches:	[hubba hubba], [mandate dated], [an annual]
Non-Matches:	[may day], [gogo], [1212]
Submitted By:	Steven Smith

Expression:	**^[a-zA-Z0-9\-\.]+\.(com|org|net|mil|edu|COM|ORG|NET|MIL|EDU)$**
Description:	Domain names: This regular expression tests the validity of a domain or hostname. It will match any valid domain name that does not contain characters which are invalid in URLs, and which ends in .com, .org, .net, .mil, or .edu. You can add additional v ...
Matches:	[3SquareBand.com], [asp.net], [army.mil]
Non-Matches:	[$SquareBand.com], [asp/dot.net], [army.military]
Submitted By:	G. Andrew Duthie